



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

Technical Paper

**Identification of stiffness properties of orthotropic lamina using
the experimental natural frequencies and mode shapes**

by

Jung-Kyu Ryou
Joshua H. Gordis

November 2007

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

Daniel T. Oliver
President

Leonard A. Ferrari
Provost

This report was prepared for Department of Mechanical and Astronautical Engineering, Naval Postgraduate School.

Reproduction of all or part of this report is authorized.

This report was prepared by:

Jung-Kyu Ryou
Visiting Professor of Mechanical & Astronautical Engineering

Joshua H. Gordis
Professor of Mechanical & Astronautical Engineering

Reviewed by:

Released by:

Anthony J. Healey
Chairman
Department of Mechanical &
Astronautical Engineering

Daniel C. Boger
Interim Associate Provost and
Dean of Research

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE November 2007	3. REPORT TYPE AND DATES COVERED Technical Report, Dec 2006 – Nov 2007	
4. TITLE AND SUBTITLE Identification of stiffness properties of orthotropic lamina using the experimental natural frequencies and mode shapes			5. FUNDING NUMBERS	
6. AUTHOR(S) Jung-Kyu Ryou and Joshua H. Gordis				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER NPS-MAE-07-003	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this report are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) <p>Mechanical properties of advanced composite lamina are identified for better mathematical modeling of composite laminate for structural analysis. Each lamina is treated as an orthotropic material under plane stress state and are assumed to be transversely isotropic. Four stiffness properties, (E_1, E_2, ν_{12}, G_{12}), are treated as design variables for minimization of a performance index. The differences between analytically obtained and experimental natural frequencies for the specimen, along with a proper weighting scheme for each mode, are minimized using the optimization routine, 'fmincon' in the MATLAB® optimization toolbox. The modal assurance criterion is utilized to construct the weighting to express the degree of correlation between mode shape vectors obtained experimentally and derived analytically.</p> <p>This study requires a series of experimental results; natural frequencies and corresponding mode shapes of the specimen. A computational tool has been developed as a result of this study. Numerical examples are investigated to demonstrate the performance of this approach. Further study with experiments may show practical benefit of current method for characterization of mechanical properties of advanced composite materials.</p>				
14. SUBJECT TERMS Mechanical Property Characterization, Composite Material, Orthotropic Lamina, Natural Frequency, Mode Shape, Optimization, Fmincon, Nine Node Mindlin Plate Element, Finite Element Method			15. NUMBER OF PAGES 93	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Mechanical properties of advanced composite lamina are identified for better mathematical modeling of composite laminate for structural analysis. Each lamina is treated as an orthotropic material under plane stress state and are assumed to be transversely isotropic. Four stiffness properties, (E_1 , E_2 , ν_{12} , G_{12}), are treated as design variables for minimization of a performance index. The differences between analytically obtained and experimental natural frequencies for the specimen, along with a proper weighting scheme for each mode, are minimized using the optimization routine, 'fmincon' in the MATLAB® optimization toolbox. The modal assurance criterion is utilized to construct the weighting to express the degree of correlation between mode shape vectors obtained experimentally and derived analytically.

This study requires a series of experimental results; natural frequencies and corresponding mode shapes of the specimen. A computational tool has been developed as a result of this study. Numerical examples are investigated to demonstrate the performance of this approach. Further study with experiments may show practical benefit of current method for characterization of mechanical properties of advanced composite materials.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION	1
II	SYSTEM MODELING	5
	A. EQUATION OF MOTION AND FINITE ELEMENT FORMULATION	5
	B. NINE-NODE REISSNER-MINDLIN PLATE ELEMENT	6
	C. LAMINATE STIFFNESS	12
	D. EIGENVALUE ANALYSIS AND SENSITIVITY OF EIGENVALUE	17
III.	OPTIMIZATION	21
	A. OVERALL PROCEDURE	21
	B. SPECIMEN DESCRIPTION	22
	C. PERFORMANCE FUNCTION AND OPTIMIZATION	23
	D. NUMERICAL EXAMPLE	24
IV.	DESCRIPTION OF COMPUTATIONAL ROUTINES	37
	A. MAIN ROUTINE	37
	B. PERFORMANCE INDEX CALCULATION	37
	C. SOLVING EIGENPROBLEM	38
	D. FINITE ELEMENT GENERATION	38
V.	CONCLUSION AND FUTURE WORKS	39
	LIST OF REFERENCES	41
	APPENDIX	43
	INITIAL DISTRIBUTION LIST	79

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Fig. 1 First order shear deformable plate -----	6
Fig. 2 Nine-node Isoparametric Element -----	7
Fig. 3 Positive rotation between material 1-2 axis (on-axis) and x-y axis -----	13
Fig. 4 z-coordinate of Lamina in a Laminate -----	16
Fig. 5 Schematic procedure to obtain mechanical properties through optimization -----	21
Fig. 6 Description of the Specimen -----	23
Fig. 7 Optimization progress, Design Variables, starting $n_{DV}=1.3*[1; 1; 1; 1]$ -----	26
Fig. 8 Optimization progress, Performance Index, starting $n_{DV}=1.3*[1; 1; 1; 1]$ -----	26
Fig. 9a Mode shape of 1 st mode with reference properties -----	27
Fig. 9b Mode shape of 2 nd mode with reference properties -----	28
Fig. 9c Mode shape of 3rd mode with reference properties -----	29
Fig. 9d Mode shape of 4 th mode with reference properties -----	30
Fig. 9e Mode shape of 5 th mode with reference properties -----	31
Fig. 10a Optimization progress, Design Variables, starting $n_{DV}=0.7*[1; 1; 1; 1]$ -----	33
Fig. 10b Optimization progress, Design Variables, starting $n_{DV}=[1.4; 0.7; 0.6; 1.2]$ -	33

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Reference mechanical Proper ties of Lamina -----	22
Table 2. Comparison of Natural Frequencies before and after Optimization (Hz) -----	25
Table 3. <i>MAC</i> 's for mode shapes with different starting vectors -----	32
Table 4. Eigenvalue sensitivities, $\partial \bar{\lambda}_j / \partial \bar{\theta}$ -----	34
Table 5. Changes of natural frequencies followed by 10% change of <i>DV</i> 's -----	34
Table 6. Eigenvalue sensitivities, $\partial \bar{\lambda}_j / \partial \bar{\theta}$ for laminate, $[\pm 45^\circ]_{2s}$ -----	35
Table 7. Eigenvalue sensitivities, $\partial \bar{\lambda}_j / \partial \bar{\theta}$ for laminate, $[0^\circ]_{8T}$ -----	35

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Based on the profound benefit of composite materials such as high specific stiffness/strength and toughness, etc., it has become common to utilize composite structures in many applications. Laminated construction is the most popular type of composite structures. Owing to many researchers' work, the advanced composite material can be mathematically modeled under well defined assumptions [1]. A lamina is the basic building block of laminated composite structures. Five independent constants in stress-strain relationship for transversely isotropic materials are reduced to four when we assume the plane stress states [2].

As material constants, the four independent mechanical properties should be determined experimentally to be used to model the composite structures mathematically. There has been extensive research to characterize these lamina properties. A series of static tests for several specimens have been proposed since 1960's [3]. A number of researchers have tried to get these properties from a dynamic test of a specimen [4~10]. It is based on the fact that the measured modal properties, e.g., natural frequencies and mode shapes, are functions of physical properties of the structure. Inverse problems have been solved to find the parameters in the mathematical model which can match the analytical modal properties with those of the real structures. As an analysis tool, various methodologies have been applied, including for example, the Rayleigh-Ritz technique [4~6] and the finite element method [7].

Most studies have focused only on the magnitude of the natural frequencies of the analytical model. The smallest (first) natural frequency of analytical model is compared with the first frequency from experiment, and so on. Natural frequencies alone, however, cannot represent satisfactorily the dynamic behavior of a system. Suppose a set of specific mechanical constants yield the natural frequencies which coincide with the corresponding experimental frequencies. We should confirm whether the analytical and experimental mode shapes match in addition. If the corresponding mode shapes of the

analytical model are similar to the experimental ones, the mechanical constants involved can be said to describe the system properly.

In this study, the four mechanical stiffness constants of a composite lamina are estimated through minimization of the performance index, which includes the similarity between the experimental mode shapes and the analytical mode shapes as well as the differences in the corresponding natural frequencies. Differences between the natural frequencies from experiment and those from analytical model for corresponding modes are weighted by factors based on the concept of modal assurance criterion [11]. The weightings for each mode express the degree of correlation between the experimental mode shapes and the analytical ones.

The analytical model of the specimen makes use of the classical laminate theory (CLT) and Reissner-Mindlin plate theory. The finite element method employs the isoparametric nine-node plate element. The specimen is a laminated plate with an arbitrary but known lay-up in a cantilever plate configuration. A proper number of finite elements is used to model the specimen. Eigenvalues and eigenvectors up to the fifth mode are calculated to compare with the experimental results. The vertical displacements at the center node of each element comprise the mode shape vector for comparison with the experimentally obtained mode shape.

Performance index minimization is performed using the optimization routine, '*fmincon.m*' in the MATLAB® optimization toolbox. Four elastic constants ($E_1, E_2, \nu_{12}, G_{12}$) have been treated as design variables for minimization. During the minimization process, the four design variables are updated such that the resulting analytical responses, i.e., natural frequencies and mode shapes, match to the corresponding experimental ones. The sensitivity of the natural frequencies with respect to the design variables is investigated.

This study requires a series of experimental results; natural frequencies and corresponding mode shapes of the specimen. A computational tool has been developed as a result of this study. All the procedures are coded using MATLAB®. Numerical examples are investigated to demonstrate the performance of this approach. Further study

with experimental results may show practical benefit of the current method for the characterization of mechanical properties of advanced composite materials.

THIS PAGE INTENTIONALLY LEFT BLANK

II. SYSTEM MODELING

A. EQUATION OF MOTION AND FINITE ELEMENT FORMULATION

The equation of motion

$$\nabla \cdot \boldsymbol{\sigma} + \vec{f} = \rho \ddot{\vec{v}} \quad \text{in } V \quad (1)$$

can be expressed as Eq. (2) by use of principle of virtual work and the divergence theorem.

$$\int \rho (\delta \vec{v})^T \ddot{\vec{v}} dV + \int (\delta \boldsymbol{\varepsilon})^T \boldsymbol{\sigma} dV = \int (\delta \vec{v})^T \vec{f} dV + \int (\delta \vec{v})^T \vec{p} dS \quad (2)$$

Here we assumed the strain-displacement relation as Eq. (3) for small strain.

$$\boldsymbol{\varepsilon} = \frac{1}{2} (\nabla \vec{v} + \nabla \vec{v}^T) \quad (3)$$

The stress-strain relation, Eq. (4) is applied to Eq. (2) to obtain the weak form of equation of motion, Eq. (5), where we deal with the case of no external forces.

$$\boldsymbol{\sigma} = D \boldsymbol{\varepsilon} \quad (4)$$

$$\int \rho (\delta \vec{v})^T \ddot{\vec{v}} dV + \int (\delta \boldsymbol{\varepsilon})^T D \boldsymbol{\varepsilon} dV = 0 \quad (5)$$

Introducing the displacement interpolation (N), the strain interpolation function (B), and the nodal displacement (U), we obtain the discretized finite element equation of motion, Eq. (6).

$$M \ddot{U} + KU = 0 \quad (6)$$

where, M is mass matrix as Eq. (7) and K is stiffness matrix as Eq. (8).

$$M = \sum_{N^e} \rho N^T N dV \quad (7)$$

$$K = \sum_{N^e} B^T D B dV \quad (8)$$

B. NINE-NODE REISSNER-MINDLIN PLATE ELEMENT (C^0)

The laminated composite plate is modeled using the nine-node Reissner-Mindlin plate elements based on the classical lamination theory [2] and the first order shear deformation theory [12]. Each node of the finite element has three degrees freedom, (ϕ_1, ϕ_2, w) , which corresponds to two rotation angles and one vertical displacement. Figure 1 shows the relationship between the shear strain and the derivative of vertical displacement in the xz -plane for the first order shear deformation theory. Transverse shear strains are expressed as follow:

$$\gamma_{xz} = \frac{\partial w}{\partial x} - \phi_1 \quad (9a)$$

$$\gamma_{yz} = \frac{\partial w}{\partial y} - \phi_2 \quad (9b)$$

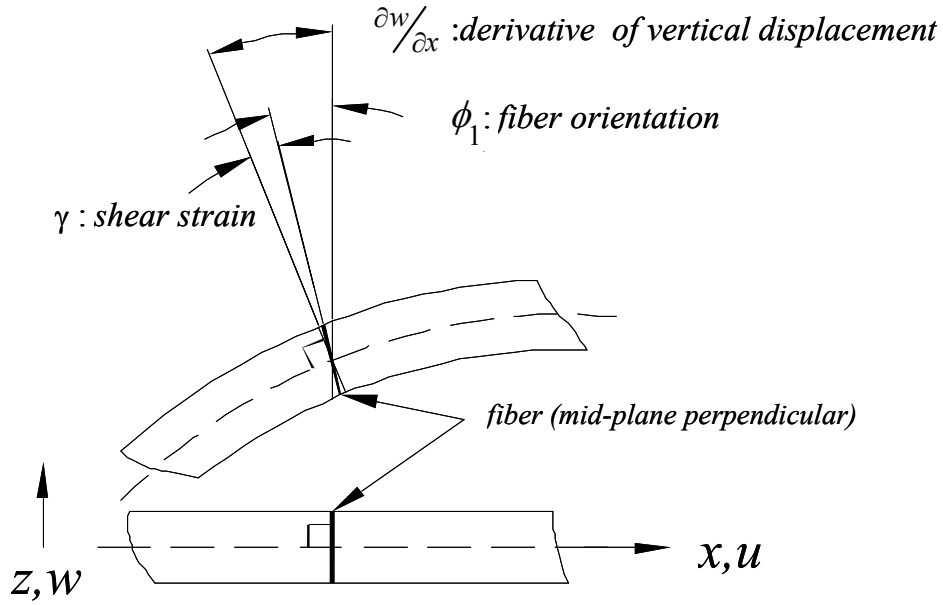


Fig. 1 First order shear deformable plate

Linear displacements along x -, y -, and z -direction, can be expressed as

$$u(x, y, z) = -z\phi_1(x, y) \quad (10a)$$

$$v(x, y, z) = -z\phi_2(x, y) \quad (10b)$$

$$w(x, y, z) = w(x, y) \quad (10c)$$

And ϕ_1 , ϕ_2 , and w are interpolated as Eqs. (11),

$$\phi_1 = \sum_{i=1}^9 h_i \phi_1^{(i)} \quad (11a)$$

$$\phi_2 = \sum_{i=1}^9 h_i \phi_2^{(i)} \quad (11b)$$

$$w = \sum_{i=1}^9 h_i w_i \quad (11c)$$

where, h_i are isoparametric interpolation shape function (Eqs. 12) for the element shown in Fig. 2 [13].

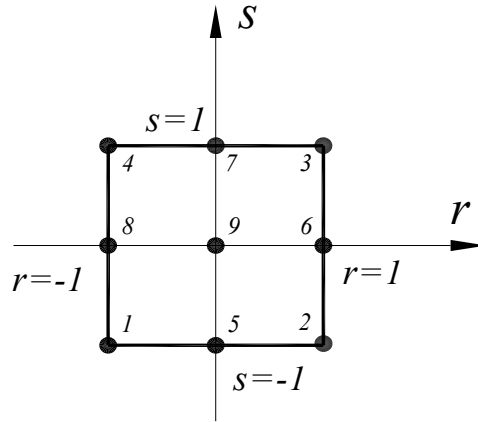


Fig. 2 Nine-node Isoparametric Element

$$h_1(r, s) = \frac{1}{4}(r^2 - r)(s^2 - s) \quad (12a)$$

$$h_2(r, s) = \frac{1}{4}(r^2 + r)(s^2 - s) \quad (12b)$$

$$h_3(r, s) = \frac{1}{4}(r^2 + r)(s^2 + s) \quad (12c)$$

$$h_4(r, s) = \frac{1}{4}(r^2 - r)(s^2 + s) \quad (12d)$$

$$h_5(r, s) = \frac{1}{2}(1 - r^2)(s^2 - s) \quad (12e)$$

$$h_6(r, s) = \frac{1}{2}(r^2 + r)(1 - s^2) \quad (12f)$$

$$h_7(r, s) = \frac{1}{2}(1 - r^2)(s^2 + s) \quad (12g)$$

$$h_8(r, s) = \frac{1}{2}(r^2 - r)(1 - s^2) \quad (12h)$$

$$h_9(r, s) = (1 - r^2)(1 - s^2) \quad (12i)$$

Displacement interpolation within a finite element is expressed in matrix form as Eq. (13).

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{bmatrix} -zh_i & 0 & 0 \\ \cdots & 0 & -zh_i & 0 & \cdots \\ 0 & 0 & -h_i \end{bmatrix} \begin{Bmatrix} \vdots \\ \phi_1^{(i)} \\ \phi_2^{(i)} \\ w^{(i)} \\ \vdots \end{Bmatrix} = N \{U^e\}, \quad i = 1 \sim 9 \quad (13)$$

The strains components of interest are expressed in terms of displacement interpolation:

$$\varepsilon_x = \frac{\partial u}{\partial x} = -z \frac{\partial \phi_1}{\partial x} = -z \sum_{i=1}^9 \frac{\partial h_i}{\partial x} \phi_1^{(i)} \quad (14a)$$

$$\varepsilon_y = \frac{\partial v}{\partial y} = -z \frac{\partial \phi_2}{\partial y} = -z \sum_{i=1}^9 \frac{\partial h_i}{\partial y} \phi_2^{(i)} \quad (14b)$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = -z \left(\frac{\partial \phi_1}{\partial y} + \frac{\partial \phi_2}{\partial x} \right) = -z \sum_{i=1}^9 \left\{ \frac{\partial h_i}{\partial y} \phi_1^{(i)} + \frac{\partial h_i}{\partial x} \phi_2^{(i)} \right\} \quad (14c)$$

$$\gamma_{xz} = \frac{\partial w}{\partial x} - \phi_1 = -\sum_{i=1}^9 h_i \phi_1^{(i)} + \sum_{i=1}^9 \frac{\partial h_i}{\partial x} w_i \quad (14d)$$

$$\gamma_{yz} = \frac{\partial w}{\partial y} - \phi_2 = -\sum_{i=1}^9 h_i \phi_2^{(i)} + \sum_{i=1}^9 \frac{\partial h_i}{\partial y} w_i \quad (14e)$$

Strain components are grouped into in-plane and transverse components as follows:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = -z \begin{bmatrix} \frac{\partial h_i}{\partial x} & 0 & 0 \\ \dots & 0 & \frac{\partial h_i}{\partial y} & 0 \dots \\ \frac{\partial h_i}{\partial y} & \frac{\partial h_i}{\partial x} & 0 \end{bmatrix} \begin{Bmatrix} \vdots \\ \phi_1^{(i)} \\ \phi_2^{(i)} \\ w^{(i)} \\ \vdots \end{Bmatrix} = -z B_I \{U^e\}, \quad i = 1 \sim 9 \quad (15a)$$

$$\begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{bmatrix} -h_i & 0 & \frac{\partial h_i}{\partial x} \\ \dots & \dots & \dots \\ 0 & -h_i & \frac{\partial h_i}{\partial y} \end{bmatrix} \begin{Bmatrix} \vdots \\ \phi_1^{(i)} \\ \phi_2^{(i)} \\ w^{(i)} \\ \vdots \end{Bmatrix} = B_S \{U^e\}, \quad i = 1 \sim 9 \quad (15b)$$

The element stiffness matrix, K^e is the sum of the bending stiffness matrix, K_B^e and the transverse shear stiffness matrix, K_S^e . The shear correction factor is assumed as $\kappa = 5/6$.

$$K^e = K_B^e + \kappa \cdot K_S^e \quad (16)$$

Detail expressions to calculate the bending stiffness and transverse stiffness are shown in Eq. (17) and (18) below:

$$\begin{aligned} K_B^e &= \iiint (-z B_I)^T \bar{Q}_I (-z B_I) dV = \iint B_I^T \left\{ \int_{-\frac{t}{2}}^{\frac{t}{2}} \bar{Q}_I z^2 dz \right\} B_I dA = \iint B_I^T D_B B_I dA \\ &= \iint \begin{bmatrix} \vdots \\ \frac{\partial h_i}{\partial x} & 0 & \frac{\partial h_i}{\partial y} \\ 0 & \frac{\partial h_i}{\partial y} & \frac{\partial h_i}{\partial x} \\ 0 & 0 & 0 \\ \vdots \end{bmatrix} \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{bmatrix} \frac{\partial h_i}{\partial x} & 0 & 0 \\ \dots & 0 & \frac{\partial h_i}{\partial y} & 0 & \dots \\ \frac{\partial h_i}{\partial y} & \frac{\partial h_i}{\partial x} & 0 \end{bmatrix} dA \end{aligned} \quad (17)$$

$$\begin{aligned}
K_S^e &= \iiint B_S^T \bar{Q}_S B_S dV = \iint B_S^T \left\{ \int_{-\frac{t}{2}}^{\frac{t}{2}} \bar{Q}_S dz \right\} B_S dA = \iint B_S^T D_S B_S dA \\
&= \iint \begin{bmatrix} \vdots \\ -h_i & 0 \\ 0 & -h_i \\ \frac{\partial h_i}{\partial x} & \frac{\partial h_i}{\partial y} \\ \vdots \end{bmatrix} \begin{bmatrix} D_{55} & D_{54} \\ D_{54} & D_{44} \end{bmatrix} \begin{bmatrix} \dots & -h_i & 0 & \frac{\partial h_i}{\partial x} & \dots \\ 0 & -h_i & \frac{\partial h_i}{\partial y} & \dots \end{bmatrix} dA
\end{aligned} \tag{18}$$

where the off-axis stiffness of lamina, \bar{Q} 's and laminate stiffness matrix, D_B and D_S will be described in detail in later section.

As the shape functions are expressed in terms of natural coordinates, we need the relationship between the physical coordinates and the natural coordinates to obtain the derivatives of shape function with respect to the physical coordinates. Using the chain rule in Eq. (19),

$$\begin{bmatrix} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = J \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \tag{19}$$

The derivatives of the shape functions with respect to the physical coordinates are expressed as Eq. (20).

$$\begin{bmatrix} \frac{\partial h_i}{\partial x} \\ \frac{\partial h_i}{\partial y} \end{bmatrix} = J^{-1} \begin{bmatrix} \frac{\partial h_i}{\partial r} \\ \frac{\partial h_i}{\partial s} \end{bmatrix} \tag{20}$$

The element mass matrix is expressed in a similar way as in Eq. (21). Mass throughout the element is assumed constant and the mass density is ρ .

$$\begin{aligned}
M^e &= \iiint \rho N^T N dV \\
&= \iiint \rho \begin{bmatrix} \vdots \\ -zh_i & 0 & 0 \\ 0 & -zh_i & 0 \\ 0 & 0 & h_i \\ \vdots \end{bmatrix} \begin{bmatrix} \dots & -zh_i & 0 & 0 & \dots \\ 0 & -zh_i & 0 & \dots \\ 0 & 0 & h_i \end{bmatrix} dV \\
&= \iint \rho \begin{bmatrix} \vdots \\ h_i & 0 & 0 \\ 0 & h_i & 0 \\ 0 & 0 & h_i \\ \vdots \end{bmatrix} \left\{ \int_{-\frac{t}{2}}^{\frac{t}{2}} \begin{bmatrix} -z \\ -z \\ 1 \end{bmatrix} [-z \quad -z \quad 1] dz \right\} \begin{bmatrix} \dots & h_i & 0 & 0 & \dots \\ 0 & h_i & 0 & \dots \\ 0 & 0 & h_i \end{bmatrix} dA \\
&= \frac{\rho t^3}{12} \iint \begin{bmatrix} \vdots \\ h_i & 0 & 0 \\ 0 & h_i & 0 \\ 0 & 0 & h_i \\ \vdots \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{12}{t^3} \end{bmatrix} \begin{bmatrix} \dots & h_i & 0 & 0 & \dots \\ 0 & h_i & 0 & \dots \\ 0 & 0 & h_i \end{bmatrix} dA \\
&= \frac{\rho t^3}{12} \iint H^T Z_{33} H dA
\end{aligned} \tag{21}$$

Gauss-Legendre quadrature is utilized to integrate the polynomials for the finite element matrix. Three points integration gives the exact value for the shape function used in this study. For the element mass matrix and the bending stiffness matrix, three by three point integration is performed. To avoid shear locking, two by two point integration is chosen for the transverse shear stiffness matrix [14].

Eqs. (22) give the expression for the matrices of the finite element.

$$K_B^e = \sum_{i=1}^3 \sum_{j=1}^3 \left\{ B_I^T D_B B_I \right\}_{(r_i, s_i)} W_i W_j \det(J) \tag{22a}$$

$$K_S^e = \sum_{i=1}^2 \sum_{j=1}^2 \left\{ B_S^T D_S B_S \right\}_{(r_i, s_i)} W_i W_j \det(J) \tag{22b}$$

$$M^e = \rho \sum_{i=1}^3 \sum_{j=1}^3 \left\{ H^T Z_{33} H \right\}_{(r_i, s_i)} W_i W_j \det(J) \tag{22c}$$

where, r_i and s_i are Gauss-Legendre integration points in the r - and s -direction, respectively; W_i and W_j are the corresponding weightings.

C. LAMINATE STIFFNESS

For transversely isotropic lamina, there are five independent coefficients as Eq. (23), where axis-1 denotes the coordinate normal to the plane of isotropy [2].

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & Q_{23} & 0 & 0 & 0 \\ & Q_{22} & Q_{23} & 0 & 0 & 0 \\ & & Q_{22} & 0 & 0 & 0 \\ & & & \frac{1}{2}(Q_{11} - Q_{12}) & 0 & 0 \\ & & & & Q_{66} & 0 \\ & & & & & Q_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{Bmatrix} \quad (23)$$

$[sym.]$

As the lamina we are dealing with can be treated as two dimensional, we introduce the plane stress state assumption ($\sigma_{33} = 0$). We express the stress-strain relationship with two separate equations, one for in-plane components and the other for transverse ones. In-plane stress-strain relations in on-axis coordinates (1-2,3-axis) are

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_6 \end{Bmatrix} = \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ & Q_{22} & 0 \\ [sym.] & & Q_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = Q_I \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} \quad (24a)$$

where, the Q_{ij} 's are expressed in terms of engineering constants in the following manner

$$Q_{11} = \frac{E_1}{1 - \nu_{12}\nu_{21}}, \quad Q_{22} = \frac{E_2}{1 - \nu_{12}\nu_{21}}, \quad Q_{12} = \frac{\nu_{12}E_2}{1 - \nu_{12}\nu_{21}}, \quad Q_{66} = G_{12} \quad (24b)$$

The transverse shear stress-strain relationship is

$$\begin{Bmatrix} \sigma_5 \\ \sigma_4 \end{Bmatrix} = \begin{Bmatrix} \tau_{13} \\ \tau_{23} \end{Bmatrix} = \begin{bmatrix} Q_{66} & 0 \\ 0 & \frac{1}{2}(Q_{11} - Q_{12}) \end{bmatrix} \begin{Bmatrix} \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = Q_S \begin{Bmatrix} \gamma_{13} \\ \gamma_{23} \end{Bmatrix} \quad (25a)$$

where, the Q_{ij} 's are expressed in engineering constants as

$$Q_{66} = G_{12}, \quad \frac{1}{2}(Q_{11} - Q_{12}) = \frac{1}{2} \cdot \frac{E_1 - E_2}{1 - \nu_{12}\nu_{21}} \quad (25b)$$

There are four independent material constants ($E_1, E_2, \nu_{12}, G_{12}$) for the case of thin lamina assumed in the plane stress state, as can be seen in Eqs. (24) and (25).

In most structural application of composite structures, the orthotropic laminae are stacked into a laminate with a certain rotation (or lamination) angle. Fig. 3 shows the relationship between the material axis ($1-2$ axis, on-axis) and the laminate axis ($x-y$ axis, off-axis).

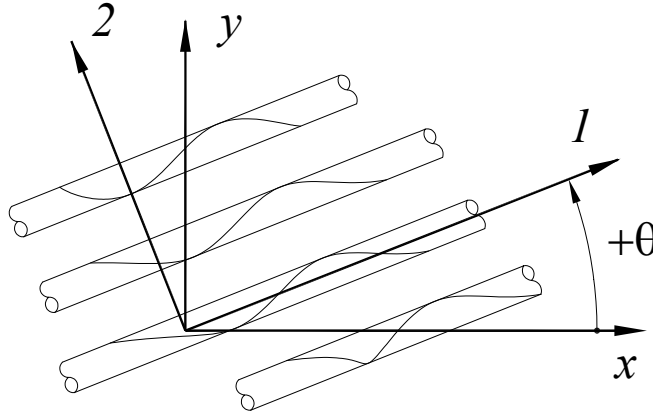


Fig. 3 Positive rotation between material $1-2$ axis (on-axis) and $x-y$ axis

In-plane stress transformation between the $1-2$ axis and $x-y$ axis is described by

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} \cos^2 \theta & \sin^2 \theta & 2\cos \theta \sin \theta \\ \sin^2 \theta & \cos^2 \theta & -2\cos \theta \sin \theta \\ -\cos \theta \sin \theta & \cos \theta \sin \theta & \cos^2 \theta - \sin^2 \theta \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = T_I \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \quad (26)$$

For the strain transformation, we must be attentive to the difference of strain tensor and engineering strain. The strain transformation is in Eq. (27).

$$\begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \gamma_{12}/2 \end{Bmatrix} = \begin{bmatrix} \cos^2 \theta & \sin^2 \theta & 2\cos \theta \sin \theta \\ \sin^2 \theta & \cos^2 \theta & -2\cos \theta \sin \theta \\ -\cos \theta \sin \theta & \cos \theta \sin \theta & \cos^2 \theta - \sin^2 \theta \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy}/2 \end{Bmatrix} = T_I \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy}/2 \end{Bmatrix} \quad (27)$$

If we introduce the scaling matrix, R ,

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad (28)$$

then,

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = R \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12}/2 \end{Bmatrix} \quad \text{and} \quad \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = R \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy}/2 \end{Bmatrix} \quad (29)$$

The off-axis stress-strain relation for in-plane components are expressed as follows:

$$\begin{aligned} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} &= T_l^{-1} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = T_l^{-1} Q_l \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} = T_l^{-1} Q_l R \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12}/2 \end{Bmatrix} = T_l^{-1} Q_l R T_l \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy}/2 \end{Bmatrix} \\ &= T_l^{-1} Q_l R T_l R^{-1} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \\ &= \bar{Q}_l \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ & \bar{Q}_{22} & \bar{Q}_{26} \\ sym. & & \bar{Q}_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \end{aligned} \quad (30)$$

where, \bar{Q}_l is the in-plane off-axis lamina stiffness matrix and its elements are calculated from the rotation angle and on-axis stiffness, Q_l 's.

$$\begin{aligned} \bar{Q}_{11} &= Q_{11} \cos^4 \theta + 2(Q_{12} + 2Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{22} \sin^4 \theta \\ \bar{Q}_{12} &= (Q_{11} + Q_{22} - 4Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{12} (\sin^4 \theta + \cos^4 \theta) \\ \bar{Q}_{22} &= Q_{11} \sin^4 \theta + 2(Q_{12} + 2Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{22} \cos^4 \theta \\ \bar{Q}_{16} &= (Q_{11} - Q_{12} - 2Q_{66}) \sin \theta \cos^3 \theta + (Q_{12} - Q_{22} + 2Q_{66}) \sin^3 \theta \cos \theta \\ \bar{Q}_{26} &= (Q_{11} - Q_{12} - 2Q_{66}) \sin^3 \theta \cos \theta + (Q_{12} - Q_{22} + 2Q_{66}) \sin \theta \cos^3 \theta \\ \bar{Q}_{66} &= (Q_{11} + Q_{22} - 2Q_{12} - 2Q_{66}) \sin^2 \theta \cos^2 \theta + Q_{66} (\sin^4 \theta + \cos^4 \theta) \end{aligned} \quad (31)$$

Off-axis transformations of transverse shear stress and strain are in Eq. (32) and (33), respectively.

$$\begin{Bmatrix} \tau_{13} \\ \tau_{23} \end{Bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{Bmatrix} \tau_{xz} \\ \tau_{yz} \end{Bmatrix} = T_S \begin{Bmatrix} \tau_{xz} \\ \tau_{yz} \end{Bmatrix} \quad (32)$$

$$\begin{Bmatrix} \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = T_S \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \quad (33)$$

Similar to the in-plane stress-strain relationship, the off-axis transverse stresses and strains are related as

$$\begin{aligned} \begin{Bmatrix} \tau_{xz} \\ \tau_{yz} \end{Bmatrix} &= T_S^{-1} \begin{Bmatrix} \tau_{13} \\ \tau_{23} \end{Bmatrix} = T_S^{-1} Q_S \begin{Bmatrix} \gamma_{13} \\ \gamma_{23} \end{Bmatrix} = T_S^{-1} Q_S T_S \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \\ &= \bar{Q}_S \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = \begin{bmatrix} \bar{Q}_{55} & \bar{Q}_{54} \\ sym. & \bar{Q}_{44} \end{bmatrix} \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \end{aligned} \quad (34)$$

where, \bar{Q}_S is the transverse shear off-axis lamina stiffness matrix and its elements are calculated from the rotation angle and on-axis stiffness, Q_S 's.

$$\begin{aligned} \bar{Q}_{55} &= Q_{66} \cos^2 \theta + \frac{1}{2} (Q_{11} - Q_{12}) \sin^2 \theta \\ \bar{Q}_{54} &= -Q_{66} \cos \theta \sin \theta + \frac{1}{2} (Q_{11} - Q_{12}) \cos \theta \sin \theta \\ \bar{Q}_{44} &= Q_{66} \sin^2 \theta + \frac{1}{2} (Q_{11} - Q_{12}) \cos^2 \theta \end{aligned} \quad (35)$$

Once several laminae are stacked to construct a laminate as shown in Fig (4), we can calculate the moment resultant by summing the stresses in each lamina. Deformation represented by the curvature $(\kappa_x, \kappa_y, \kappa_{xy})$ is related to the moment resultant (M_x, M_y, M_{xy}) through the bending stiffness of laminate, D_B , as in Eq. (36).

$$\begin{aligned} \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} &= \int_{-\frac{t}{2}}^{\frac{t}{2}} z \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} dz = - \int_{-\frac{t}{2}}^{\frac{t}{2}} z \bar{Q}_I \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} dz = \int_{-\frac{t}{2}}^{\frac{t}{2}} z^2 \bar{Q}_I \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} dz \\ &= \sum_{k=1}^N \bar{Q}_I \Big|_k \int_{z_k}^{z_{k+1}} z^2 dz \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} = \frac{1}{3} \sum_{k=1}^N \bar{Q}_I \Big|_k (z_{k+1}^3 - z_k^3) \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} = D_B \begin{Bmatrix} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{Bmatrix} \end{aligned} \quad (36)$$

In a similar way, the transverse shear force resultant (Q_{xz}, Q_{yz}) is expressed in terms of the transverse shear strain $(\gamma_{xz}, \gamma_{yz})$ in the laminate with the transverse shear stiffness, D_S , which is shown in Eq. (37).

$$\begin{aligned} \begin{Bmatrix} Q_{xz} \\ Q_{yz} \end{Bmatrix} &= \int_{-\frac{t}{2}}^{\frac{t}{2}} \begin{Bmatrix} \tau_{xz} \\ \tau_{yz} \end{Bmatrix} dz = - \int_{-\frac{t}{2}}^{\frac{t}{2}} \bar{Q}_S \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} dz = \sum_{k=1}^N \bar{Q}_S \Big|_k \int_{z_k}^{z_{k+1}} dz \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \\ &= \sum_{k=1}^N \bar{Q}_S \Big|_k (z_{k+1} - z_k) \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} = D_S \begin{Bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{Bmatrix} \end{aligned} \quad (37)$$

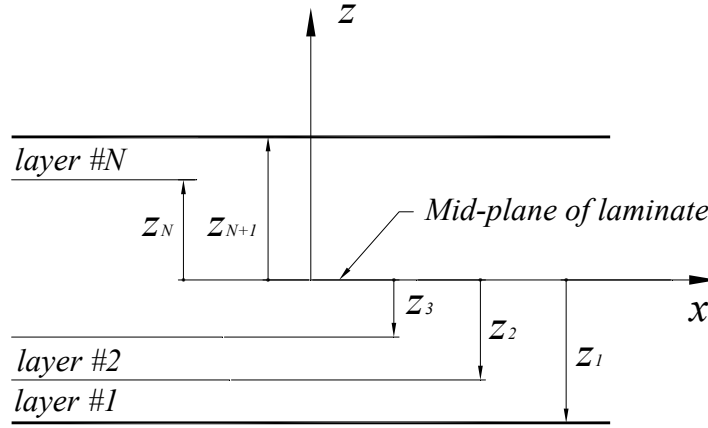


Fig. 4 z-coordinate of Lamina in a Laminate

D. EIGENVALUE ANALYSIS AND SENSITIVITY OF EIGENVALUE

Assuming that the displacement response is harmonic, the equation of motion in Eq. (6) can be expressed as Eq. (38), the so called structural eigenproblem.

$$[K - \lambda M] \{\phi\} = \{0\} \quad (38)$$

where, λ_j and ϕ_j are j -th eigenvalue and eigenvector, respectively. The eigenvector is normalized with respect to mass matrix. The mass and stiffness matrices in Eq. (38) are obtained by assembling the element matrices in Eqs. (16) and (22), and boundary conditions are applied. The lowest five eigenvalues and corresponding eigenvectors are

calculated. They are used to construct the performance function for optimization together with the experimental natural frequencies and mode shapes.

As the four mechanical constants, $E_1, E_2, \nu_{12}, G_{12}$, are treated as design variables for optimization, it is meaningful to investigate the sensitivity of natural frequencies with respect to these design variables [15]. Differentiating the eigensystem, Eq. (38), with respect to a design variable, θ , yields

$$\frac{\partial K}{\partial \theta} \phi_j + K \frac{\partial \phi_j}{\partial \theta} = \frac{\partial \lambda_j}{\partial \theta} M \phi_j + \lambda_j \frac{\partial M}{\partial \theta} \phi_j + \lambda_j M \frac{\partial \phi_j}{\partial \theta} \quad (39)$$

Pre-multiplying ϕ_j^T to above Eq. (39) gives us

$$\phi_j^T \left(\frac{\partial K}{\partial \theta} - \frac{\partial \lambda_j}{\partial \theta} M - \lambda_j \frac{\partial M}{\partial \theta} \right) \phi_j + \phi_j^T (K - \lambda_j M) \frac{\partial \phi_j}{\partial \theta} = 0 \quad (40)$$

If we note that the relationships ($\phi_j^T M \phi_j = 1$; K and M are symmetric; and $(K - \lambda_j M) \phi_j = 0$), Eq. (40) reduces to

$$\frac{\partial \lambda_j}{\partial \theta} = \phi_j^T \frac{\partial K}{\partial \theta} \phi_j \quad (41)$$

for the case of $\frac{\partial M}{\partial \theta} = 0$,

The normalized sensitivity of the eigenvalue is defined as Eq. (42):

$$\frac{\partial \bar{\lambda}_j}{\partial \theta} = \frac{1}{\lambda_j} \left(\frac{\partial \lambda_j}{\partial \theta} \right) \quad (42)$$

We are interested in the normalized eigenvalues and its sensitivity with respect to the normalized design variable, $\bar{\theta} = \theta / \theta_0$.

$$\frac{\partial \bar{\lambda}_j}{\partial \bar{\theta}} = \frac{\partial \bar{\lambda}_j}{\partial \theta} \frac{\partial \theta}{\partial \bar{\theta}} = \frac{\partial \bar{\lambda}_j}{\partial \theta} \theta_0 = \frac{1}{\lambda_j} \left(\phi_j^T \frac{\partial K}{\partial \theta} \phi_j \right) \theta_0 \quad (43)$$

Eqs. (16)~(18) gives us

$$\frac{\partial K^e}{\partial \theta} = \iint B_I^T \frac{\partial D_B}{\partial \theta} B_I dA + \iint B_S^T \frac{\partial D_S}{\partial \theta} B_S dA \quad (44)$$

The normalized eigenvalue sensitivity with respect to the normalized design variables depends on the eigenvectors and the derivatives of stiffness matrices, $\frac{\partial D_B}{\partial \theta}$ and $\frac{\partial D_S}{\partial \theta}$.

According to the description for the laminate stiffness, we can see that derivatives of stiffness matrices are expressed eventually in term of derivatives of the on-axis stiffness,

$\frac{\partial Q_{ij}}{\partial \theta}$ given as follows:

$$\frac{\partial Q_{11}}{\partial E_1} = \frac{1}{1 - \nu_{12}\nu_{21}} - \frac{\nu_{12}\nu_{21}}{(1 - \nu_{12}\nu_{21})^2} \quad (45a)$$

$$\frac{\partial Q_{22}}{\partial E_1} = -\frac{\nu_{21}^2}{(1 - \nu_{12}\nu_{21})^2} \quad (45b)$$

$$\frac{\partial Q_{12}}{\partial E_1} = -\frac{\nu_{12}\nu_{21}^2}{(1 - \nu_{12}\nu_{21})^2} \quad (45c)$$

$$\frac{\partial Q_{66}}{\partial E_1} = 0 \quad (45d)$$

$$\frac{\partial Q_{11}}{\partial E_2} = \frac{\nu_{12}^2}{(1 - \nu_{12}\nu_{21})^2} \quad (46a)$$

$$\frac{\partial Q_{22}}{\partial E_2} = \frac{1}{1 - \nu_{12}\nu_{21}} + \frac{\nu_{12}\nu_{21}}{(1 - \nu_{12}\nu_{21})^2} \quad (46b)$$

$$\frac{\partial Q_{12}}{\partial E_2} = \frac{\nu_{12}}{1 - \nu_{12}\nu_{21}} + \frac{\nu_{12}\nu_{21}^2}{(1 - \nu_{12}\nu_{21})^2} \quad (46c)$$

$$\frac{\partial Q_{66}}{\partial E_2} = 0 \quad (46d)$$

$$\frac{\partial Q_{11}}{\partial \nu_{12}} = \frac{2\nu_{12}E_2}{(1-\nu_{12}\nu_{21})^2} \quad (47a)$$

$$\frac{\partial Q_{22}}{\partial \nu_{12}} = \frac{2\nu_{21}E_2}{(1-\nu_{12}\nu_{21})^2} \quad (47b)$$

$$\frac{\partial Q_{12}}{\partial \nu_{12}} = \frac{E_2}{1-\nu_{12}\nu_{21}} + \frac{2E_2\nu_{12}\nu_{21}}{(1-\nu_{12}\nu_{21})^2} \quad (47c)$$

$$\frac{\partial Q_{66}}{\partial \nu_{12}} = 0 \quad (47d)$$

$$\frac{\partial Q_{11}}{\partial G_{12}} = 0 \quad (48a)$$

$$\frac{\partial Q_{22}}{\partial G_{12}} = 0 \quad (48b)$$

$$\frac{\partial Q_{12}}{\partial G_{12}} = 0 \quad (48c)$$

$$\frac{\partial Q_{66}}{\partial G_{12}} = 1 \quad (48d)$$

THIS PAGE INTENTIONALLY LEFT BLANK

III. OPTIMIZATION

A. OVERALL PROCEDURE

Overall procedure for obtaining the mechanical property constants is shown in the Fig. 5. Four mechanical constants are treated as design variables for optimization. Arbitrary initial values are assumed for the design variables. Modal parameters, which are the natural frequencies and mode shapes in this study, are calculated with these initial values for the system model of the specimen.

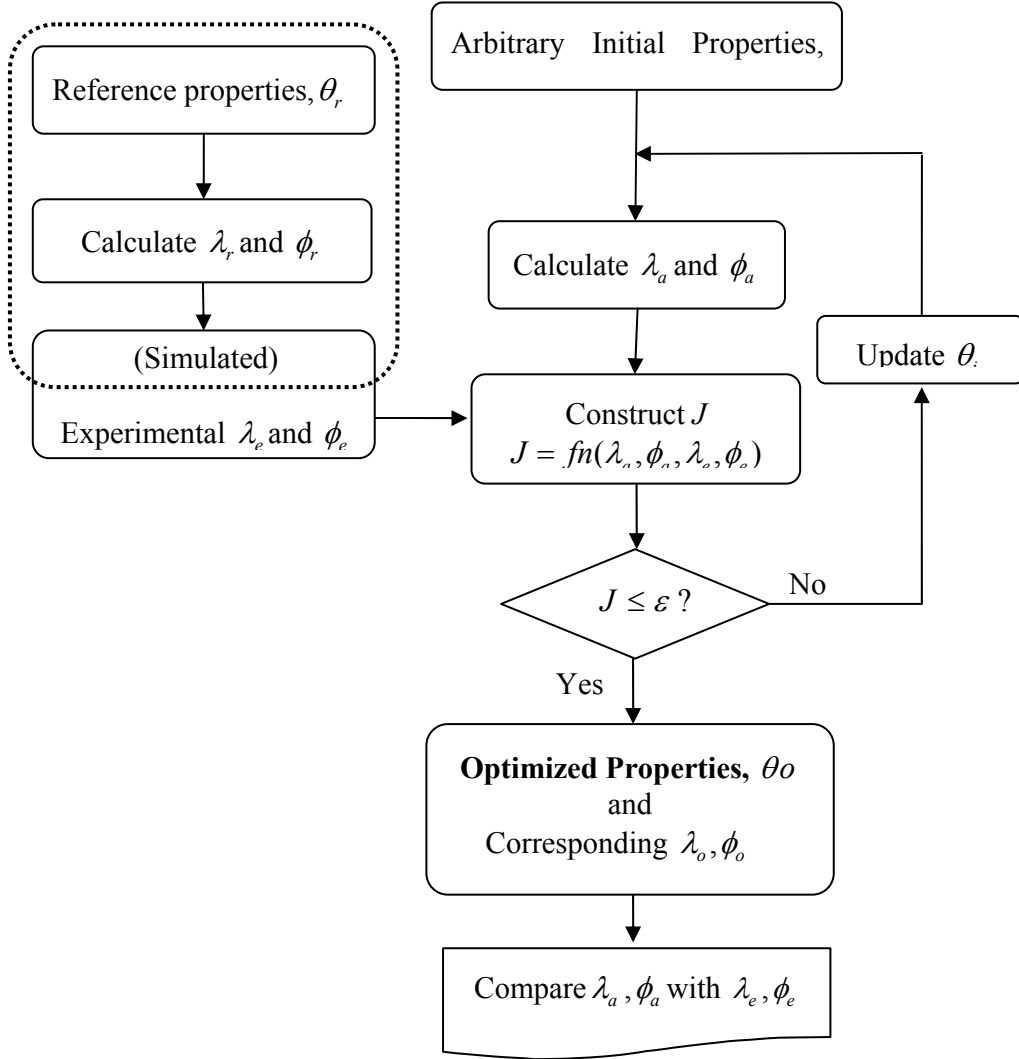


Fig. 5 Schematic procedure to obtain mechanical properties through optimization

These modal parameters are combined with the experimental ones to yield the performance index to be minimized. Until satisfactory minimization of performance index is obtained, the design variables are updated repeatedly. Natural frequencies and mode shapes are calculated at every iteration. Once optimization reaches the goal, the natural frequencies of the mathematical model are close enough to the experimental results. The design variables at this step are taken as the desired mechanical properties of the lamina.

It is required to have experimental modal parameters to make the performance function for optimization. At present, this study is primarily concerned with building a computational tool for mechanical property identification. The experimental data are simulated, and were generated from the analysis results with reference mechanical properties. This step shall be replaced when experimental data are available.

B. SPECIMEN DESCRIPTION

In this study, the system model is a laminated plate in cantilever configuration. Its planform is as shown in Fig. 6. The plate is modeled with a nine-node isoparametric element based on the first order shear deformable Reissner-Mindlin plate theory. The laminate will be constructed with orthotropic lamina whose mechanical properties are to be identified. The lay-up angle of each lamina are specified and the density and ply thickness are assumed known. Table 1 shows the reference properties for graphite/epoxy lamina. These properties would be obtained through a series of static coupon tests.

Table 1. Reference mechanical Properties of Lamina

E_1	122.5	<i>GPa</i>
E_2	7.929	<i>GPa</i>
ν_{12}	0.329	-
G_{12}	3.585	<i>GPa</i>
Thickness per ply	0.15	<i>mm</i>
Density of lamina	1500	<i>kg/m³</i>

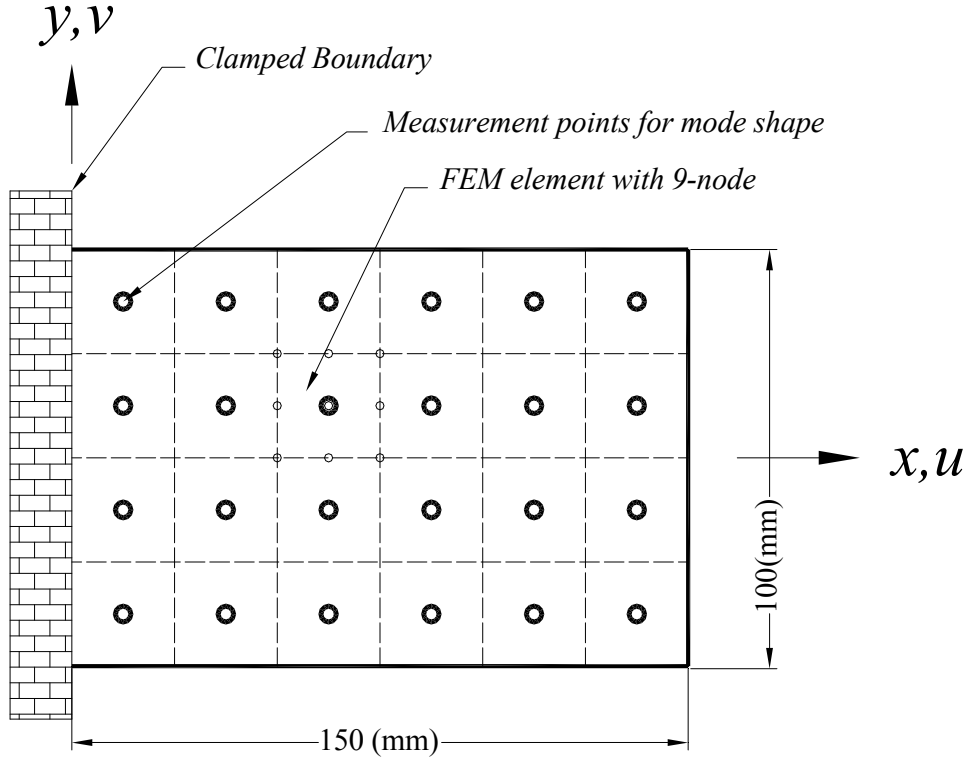


Fig. 6 Description of the Specimen

C. PERFORMANCE FUNCTION AND OPTIMIZATION

The analytical modal parameters are used, together with the experimental modal parameters, to construct the performance function to be minimized. Design variables are updated until the performance index becomes sufficiently small. Optimization scheme, ‘*fmincon* in MATLAB’, uses a sequential quadratic programming method and BFGS formula to update an estimate of the Hessian. Details of the scheme can be found in references [16, 17].

Restrictions on the engineering constants (Eq. 49) which come from elastomechanical constraints are handled as inequality constraints in *fmincon*.

$$|\nu_{12}| \leq \sqrt{\frac{E_1}{E_2}} \quad (49)$$

It is important to have a physically reasonable performance function for this kind of problem. The mode shape for i^{th} mode obtained experimentally is denoted as $\phi_e^{(i)}$ and corresponding analytical one is $\phi_a^{(i)}$. The closeness of these two vectors can be represented by the angle, θ_i , between them. It is known as the MAC_i .

$$MAC_i = \frac{|\phi_a^T \phi_e|^2}{(\phi_a^T \phi_a)(\phi_e^T \phi_e)} = \cos^2 \theta_i \quad (50)$$

In the case that a specific mode shape calculated is quite different from one obtained in experiment, it is of no use to try to match the natural frequencies from analysis and experiment. Therefore the frequency differences for each mode are weighted with MAC 's.

$$J = \sum_{i=1}^{N_m} (MAC_i) \cdot \sqrt{\left(\frac{f_a^{(i)} - f_e^{(i)}}{f_e^{(i)}} \right)^2} \quad (51)$$

where, $f_a^{(i)}$ and $f_e^{(i)}$ are natural frequencies from analysis and experiment, respectively.

N_m is number of modes in consideration and five in this study.

D. NUMERICAL EXAMPLE

As a numerical example, optimization procedure and result are demonstrated for a laminate whose stacking sequence is $[\pm 45^\circ / 0^\circ_2 / 90^\circ]_s$. The specimen configuration is as in Fig. 6. Reference properties are as in Table 1. Modal data calculated with these properties are used as simulated experimental data. A starting vector of mechanical constants is chosen arbitrarily. In this example, it is given as follows:

$$\{n_{-DV}\} = \left\{ \frac{E_1}{E_1^R}, \frac{E_2}{E_2^R}, \frac{\nu_{12}}{\nu_{12}^R}, \frac{G_{12}}{G_{12}^R} \right\} = [1.3; 1.3; 1.3; 1.3]$$

where, superscript R denotes reference value as in Table 1. Lower and upper bounds of the normalized design variables were 0.5 and 1.5, respectively. Table 2 shows natural frequencies with optimized mechanical properties together with the reference natural

frequencies. Here the reference natural frequencies are treated as experimental ones. Changes in design variables in the course of optimization are shown in Fig. 7 and the performance index history is in Fig. 8.

Optimization leads the natural frequencies of the specimen to the target value satisfactorily. Table 2 shows that the natural frequency for each mode reaches the corresponding experimental value. As we do not have in this study any error or noise that usually exists in real experiments, we obtained good results which are almost the same as the simulated numerical frequencies. Fig. 9 shows the mode shapes obtained with the optimized DV 's.

Table 2. Comparison of Natural Frequencies before and after Optimization (Hz)

Mod	With Starting DV 's	With Optimized DV 's	Experiment (Simulated)
1 st	70.12 (114.4%)	→	61.32 (100%)
2 nd	270.57 (113.8%)	→	237.86 (100%)
3 rd	423.30 (114.2%)	→	370.71 (100%)
4 th	866.43 (114.0%)	→	759.91 (100%)
5 th	939.54 (114.1%)	→	823.18 (100%)

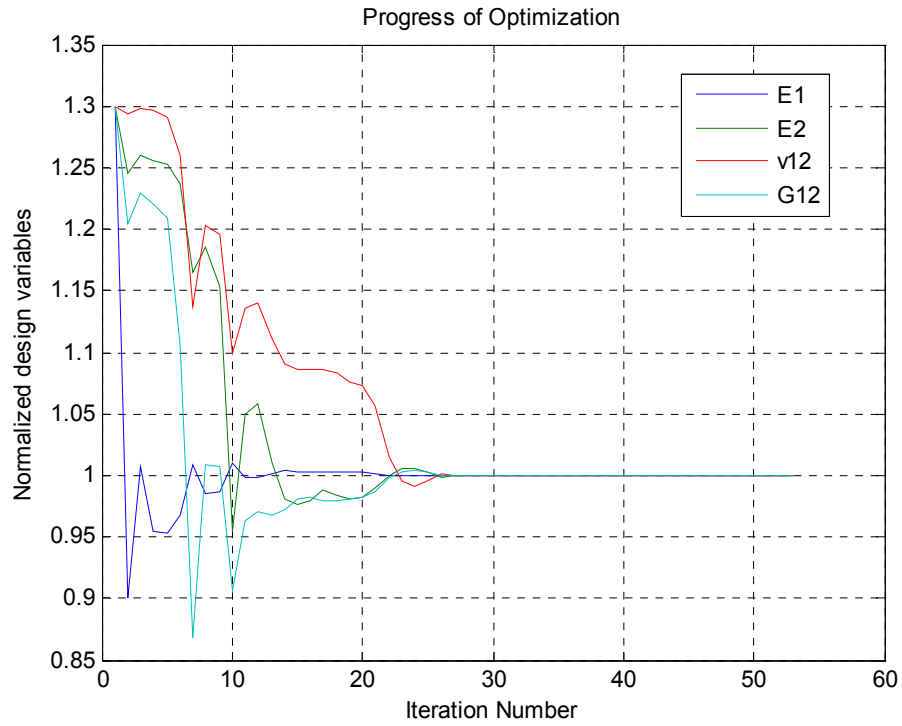


Fig. 7 Optimization progress, Design Variables, starting $n_{DV}=1.3*[1; 1; 1; 1]$

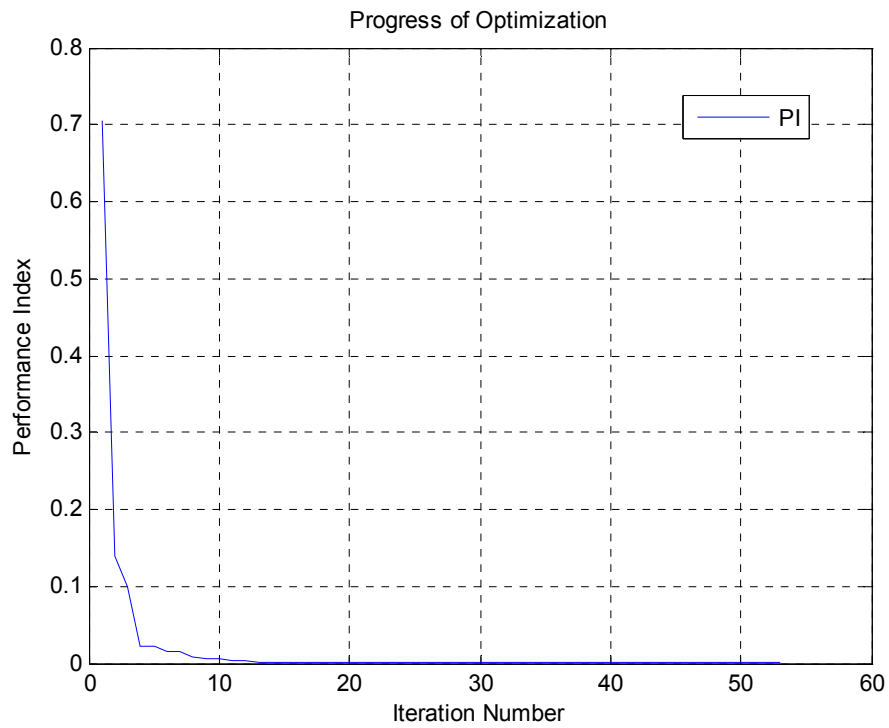


Fig. 8 Optimization progress, Performance Index, starting $n_{DV}=1.3*[1; 1; 1; 1]$

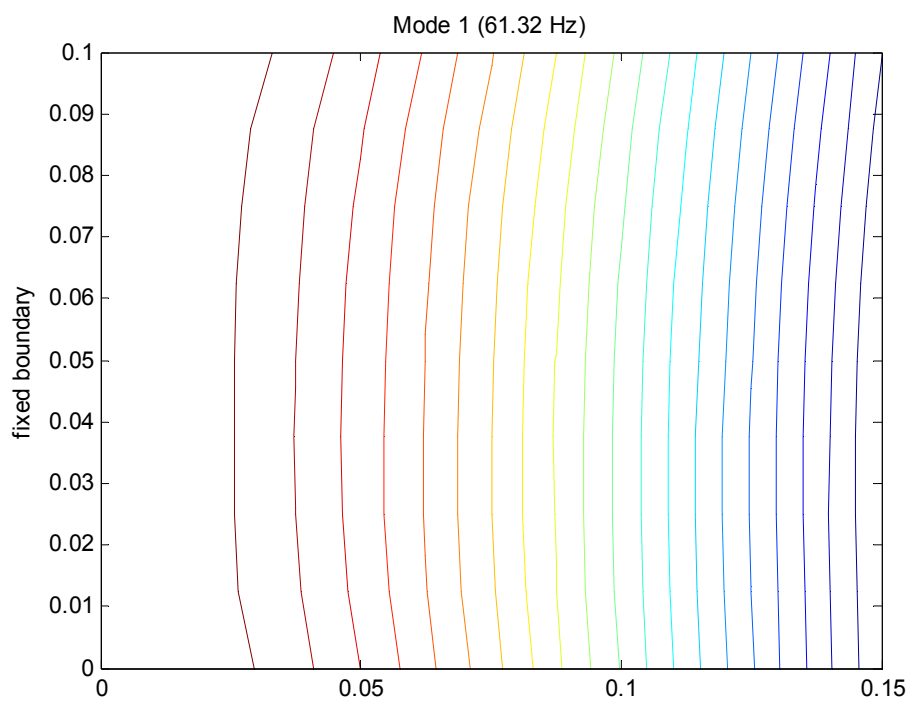
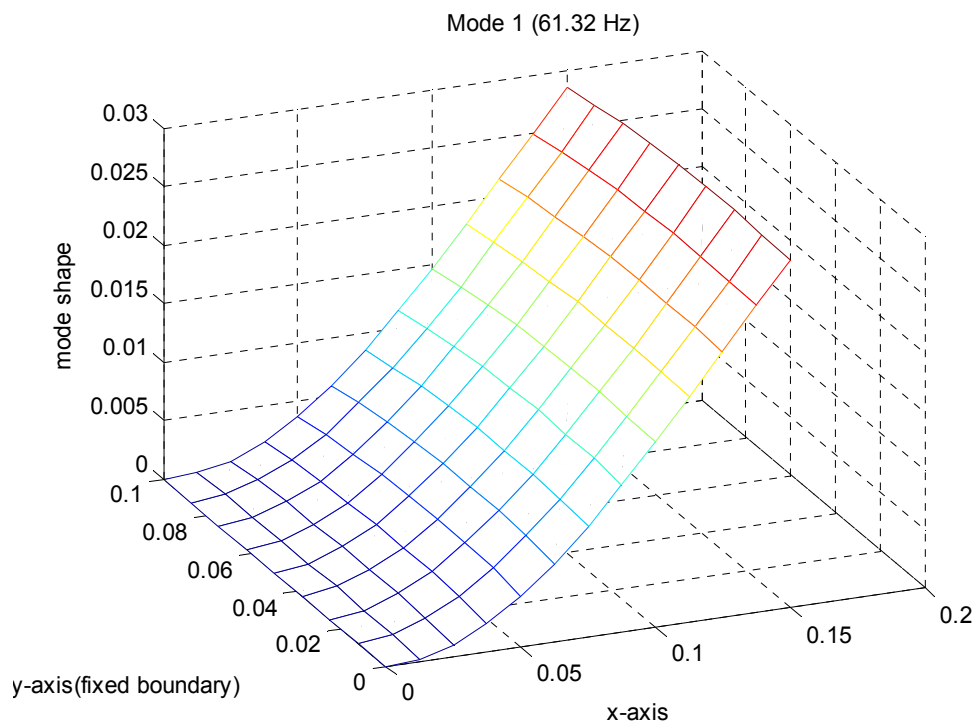


Fig. 9a Mode shape of 1st mode with reference properties

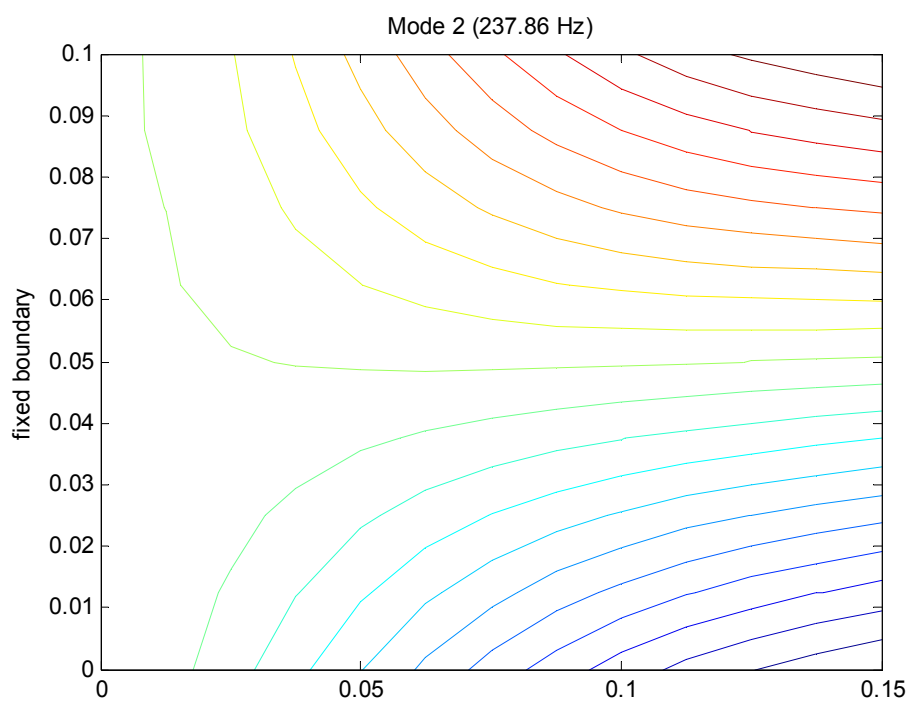
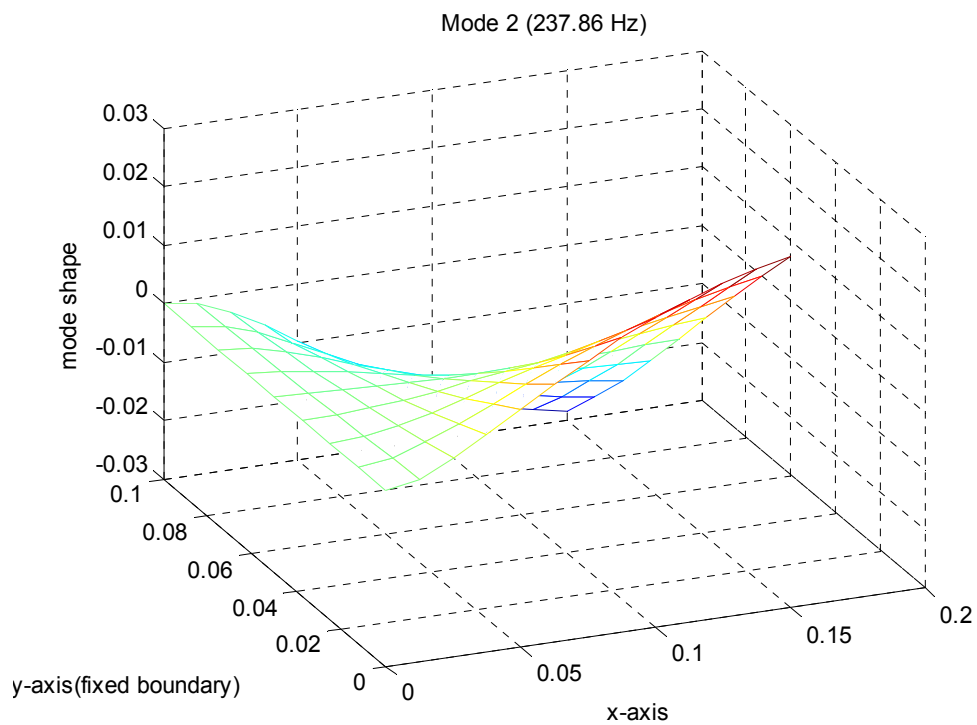


Fig. 9b Mode shape of 2nd mode with reference properties

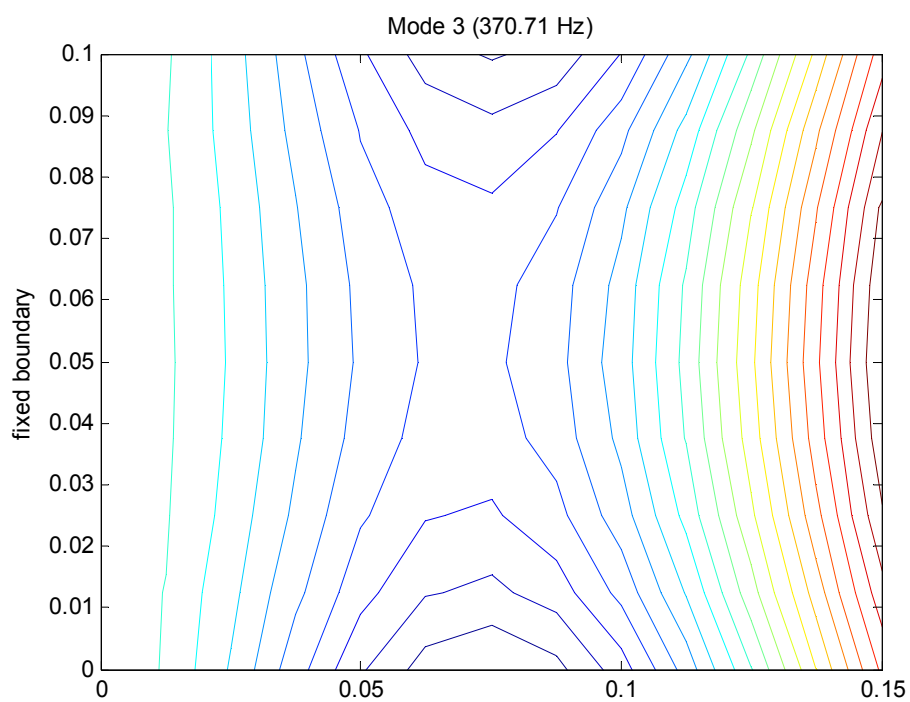
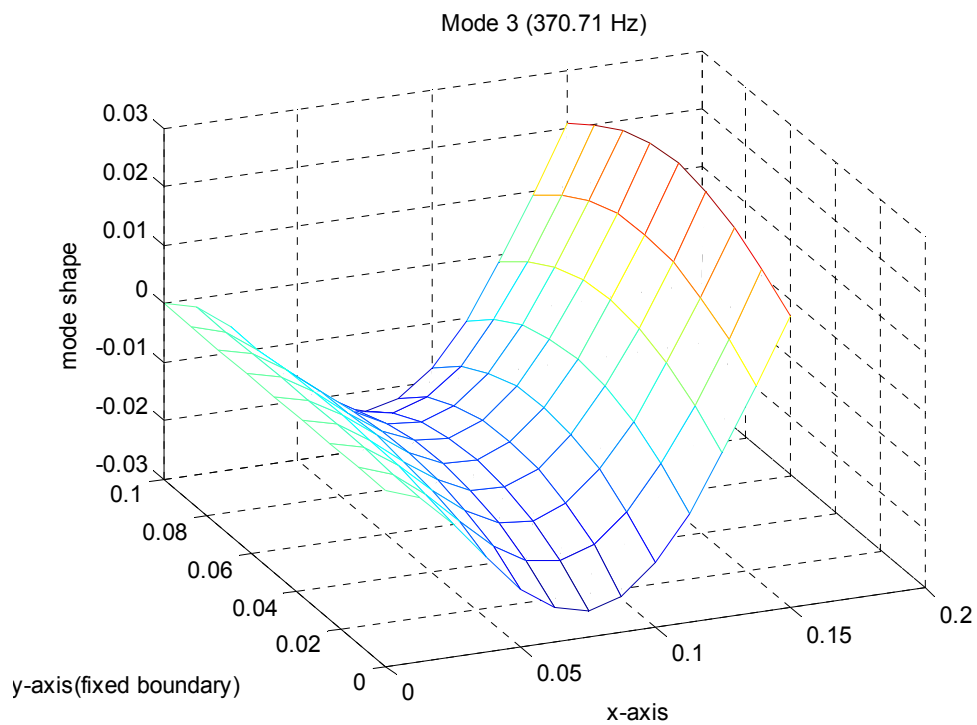


Fig. 9c Mode shape of 3rd mode with reference properties

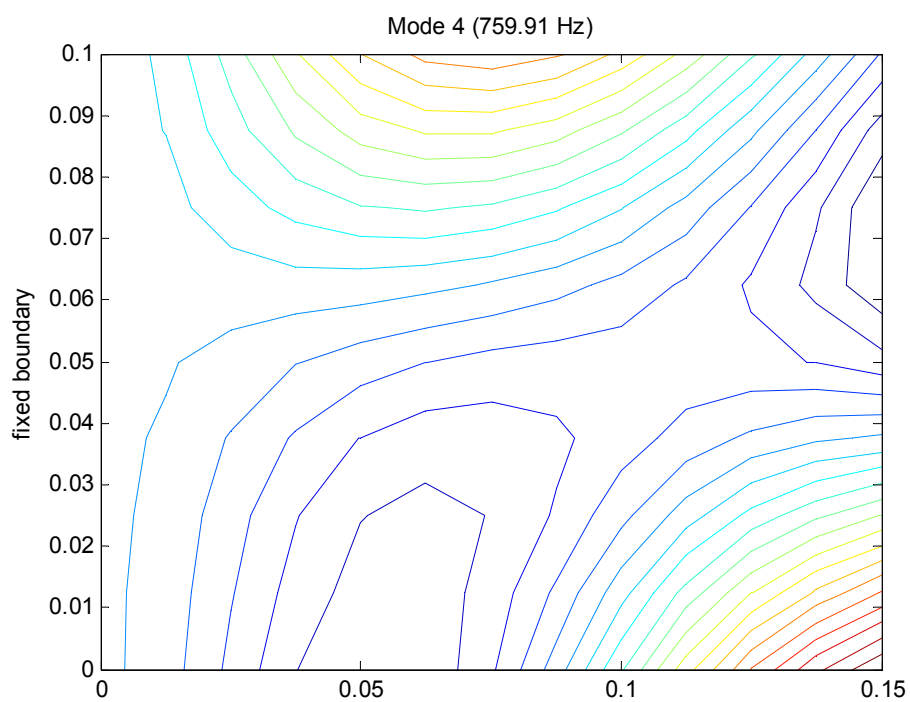
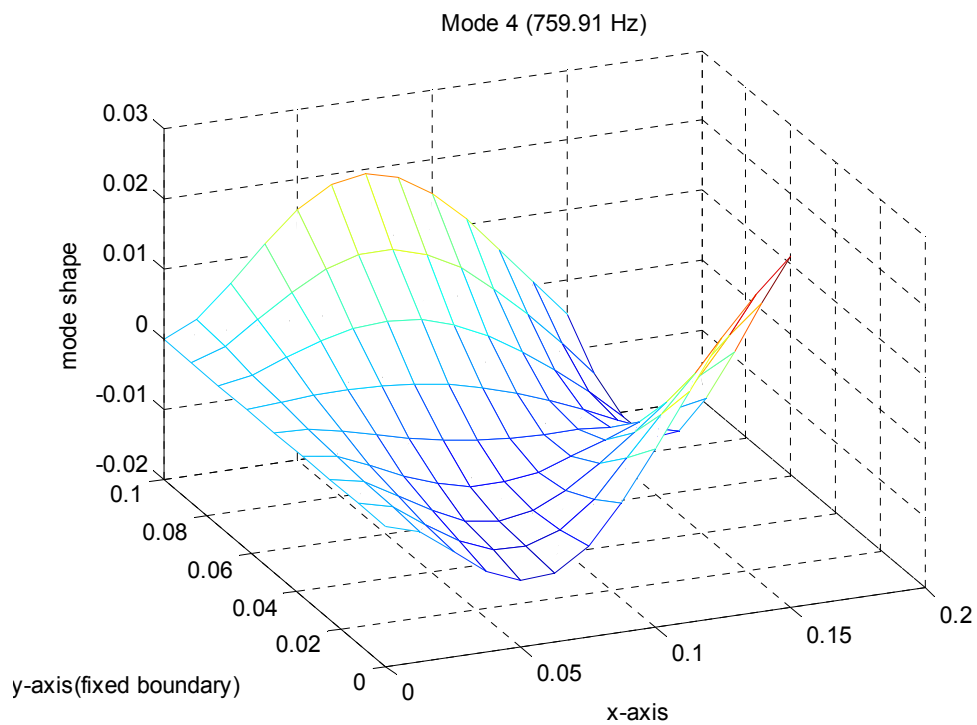


Fig. 9d Mode shape of 4th mode with reference properties

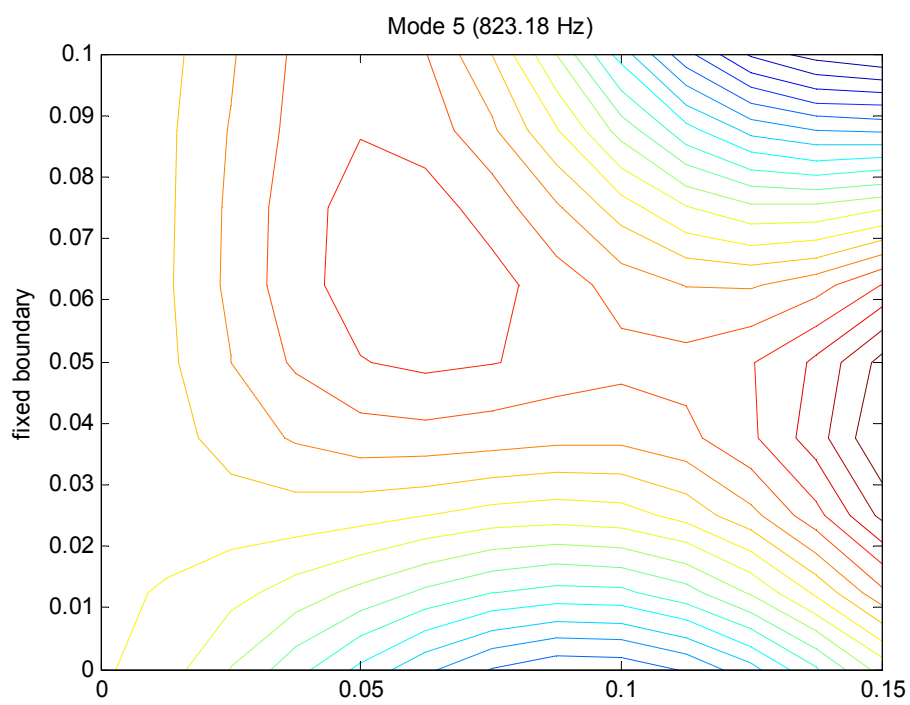
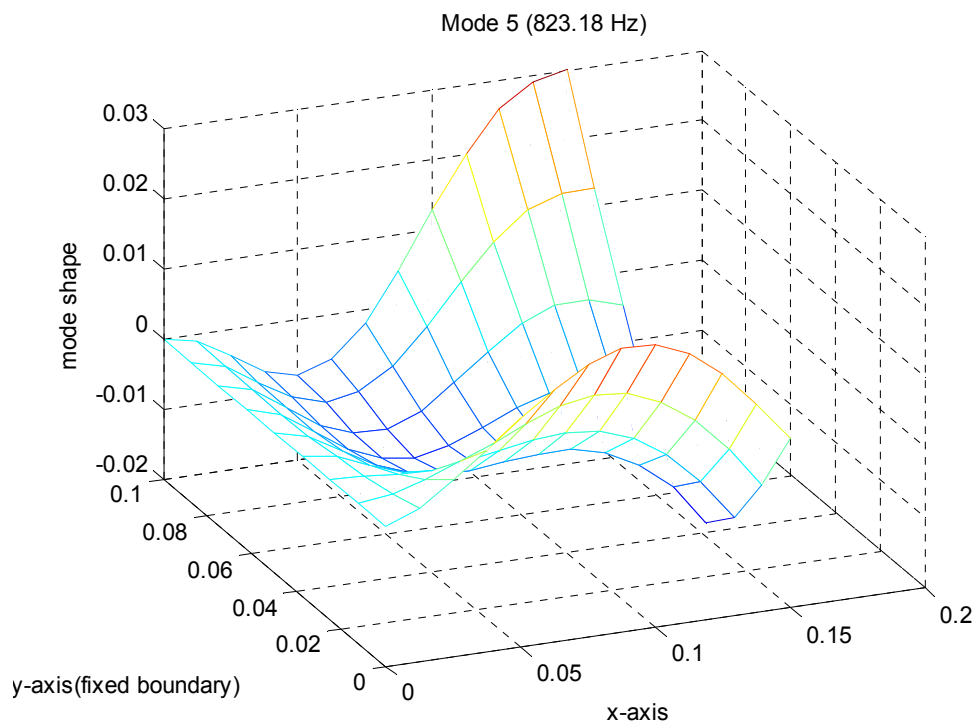


Fig. 9e Mode shape of 5th mode with reference properties

To see the situation with different starting points during optimization, two additional starting points are chosen. Figs. 10a and 10b show the progress of optimization in terms of design variables. For the case of Fig. 10a, the starting vector of design variables is $[0.7; 0.7; 0.7; 0.7]$, i.e., 70% of the reference values. Optimization yields the values of $[0.9999; 1.0005; 0.9982; 1.0006]$. If we start with the vector, $[1.4; 0.7; 0.6; 1.2]$, we reach the result, $[1.0000; 0.9998; 1.0011; 0.9997]$. It cannot be mentioned in general that any starting vector may give the desired result, $[1.0000; 1.0000; 1.0000; 1.0000]$. Concerning the initial estimate of the mechanical properties, which is necessary for the approach in this study, we have adequate freedom as needed to choose the starting vector.

The weighting factor, MAC , in the performance function plays a role to reduce the effect of specific mode of which the mode shape from the mathematical model differs from that of the experiments. The experimental mode shape vector in this study is constructed with the z -displacement at the center node of finite elements. As mention before, it is obtained not from experiment but numerically. Table 3 shows the starting and optimized vector of design variables and the MAC 's for the mode shape with several starting vectors. They are all compared with the mode shapes with reference properties which are treated as experimental ones in this study.

Table 3 MAC 's for mode shapes with different starting vectors

	Case 1	Case 2	Case 3
Starting vector ↓ Optimized vector	$\begin{Bmatrix} 1.3 \\ 1.3 \\ 1.3 \\ 1.3 \end{Bmatrix} \rightarrow \begin{Bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 1.0 \end{Bmatrix}$	$\begin{Bmatrix} 0.7 \\ 0.7 \\ 0.7 \\ 0.7 \end{Bmatrix} \rightarrow \begin{Bmatrix} 0.9999 \\ 1.0005 \\ 0.9982 \\ 1.0006 \end{Bmatrix}$	$\begin{Bmatrix} 1.4 \\ 0.7 \\ 0.6 \\ 1.2 \end{Bmatrix} \rightarrow \begin{Bmatrix} 1.0000 \\ 0.9998 \\ 1.0011 \\ 0.9997 \end{Bmatrix}$
MAC of Starting vector	$\begin{Bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 0.9996 \\ 0.9995 \end{Bmatrix}$	$\begin{Bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 0.9995 \\ 0.9995 \end{Bmatrix}$	$\begin{Bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 0.9827 \\ 0.9818 \end{Bmatrix}$

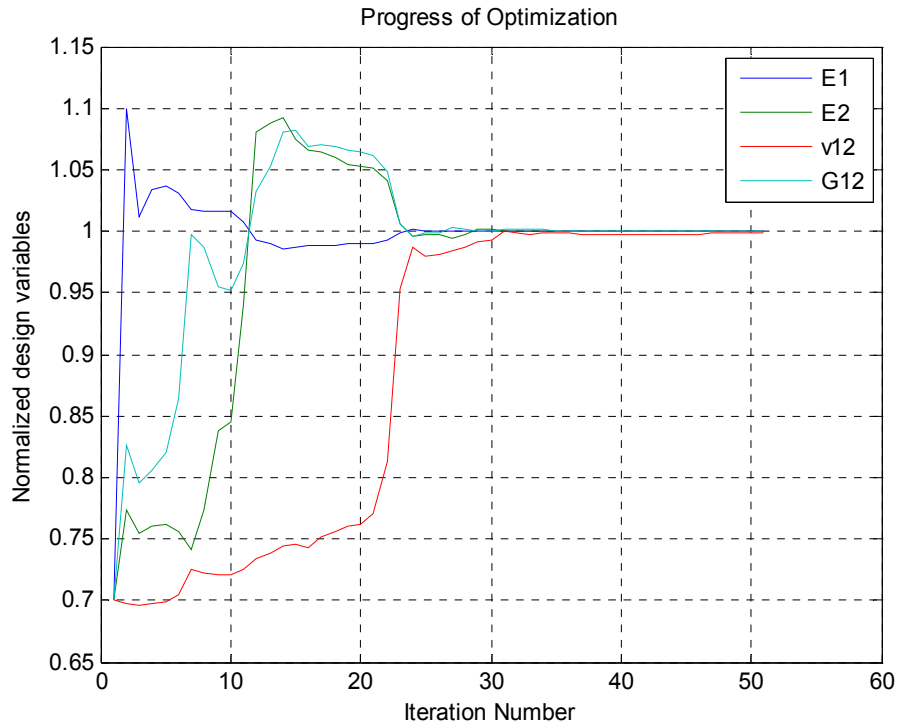


Fig. 10a Optimization progress, Design Variables, starting $n_{DV}=0.7*[1; 1; 1; 1]$

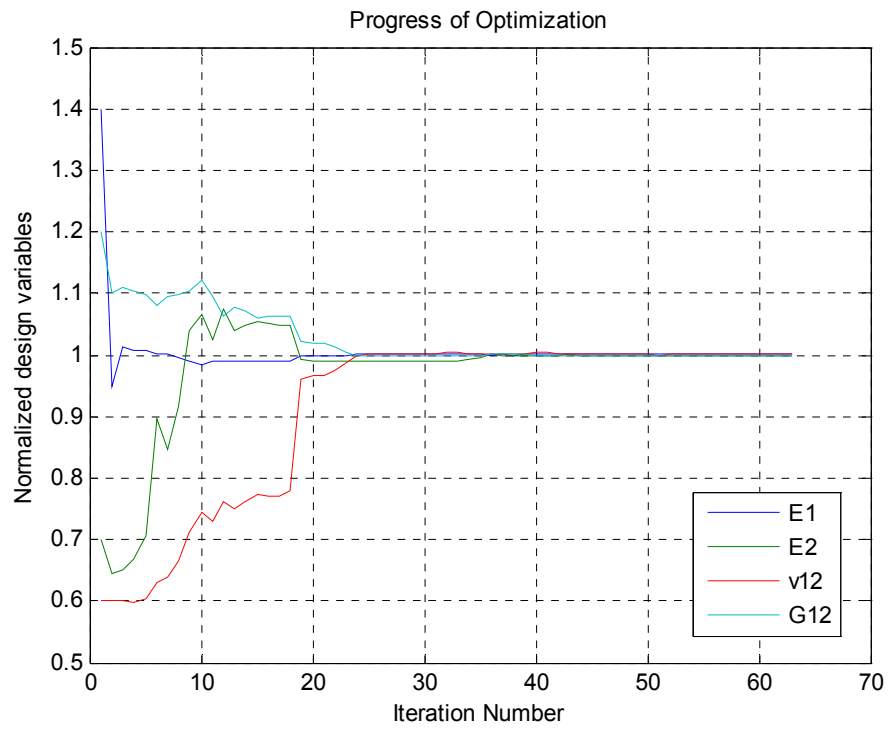


Fig. 10b Optimization progress, Design Variables, starting $n_{DV}=[1.4; 0.7; 0.6; 1.2]$

This study proposes a methodology for identifying mechanical properties of orthotropic lamina. It is therefore of use to investigate the sensitivity of eigenvalues or natural frequencies of the specimen with respect to the mechanical properties. Table 4 shows them using Eq. (43) through (48). As the laminate in this example is constructed with many layers and the lamination angles are $\pm 45^\circ$, 0° , and 90° , the stiffness in fiber direction, E_1 , has prevailing influence on all modes in consideration. Sensitivity over 0.1 is shaded in the table. G_{12} has the sensitivity over 0.1 only for 1st and 3rd modes.

Table 4. Eigenvalue sensitivities, $\partial \bar{\lambda}_j / \partial \bar{\theta}$

	1 st mode	2 nd mode	3 rd mode	4 th mode	5 th mode
E_1	0.8589	0.9221	0.8338	0.8627	0.8436
E_2	0.0403	0.0393	0.0327	0.0716	0.0633
ν_{12}	0.0175	-0.0174	0.0083	-0.0015	0.0049
G_{12}	0.1008	0.0386	0.1334	0.0656	0.0931

With the aid of Eq. (52), we can express the percent of changes in natural frequencies according to the changes of design variables.

$$\frac{f_{new}}{f_{old}} = \sqrt{1 + \frac{d\bar{\lambda}}{d\bar{\theta}} \cdot (\Delta\bar{\theta})} \quad (52)$$

Table 5 shows the percent changes of natural frequencies when each design variable changes 10%. If we change E_1 by 10%, the natural frequencies up to 5th mode changes by approximately 4%. Other design variables, however, seem not to have noticeable effect on the natural frequencies compare to E_1 .

Table 5. Changes of natural frequencies followed by 10% change of DV 's

	1 st mode	2 nd mode	3 rd mode	4 th mode	5 th mode
$\Delta E_1 = 10\%$	4.21 %	4.51 %	4.09 %	4.22 %	4.13 %
$\Delta E_2 = 10\%$	0.20 %	0.20 %	0.16 %	0.36 %	0.32 %
$\Delta \nu_{12} = 10\%$	0.09 %	-0.09 %	0.04 %	-0.01 %	0.02 %
$\Delta G_{12} = 10\%$	0.50 %	0.19 %	0.66 %	0.33 %	0.46 %

The major influence of E_I on the natural frequencies is mentioned for the laminate of the previous example. For a laminate which is constructed with laminae in special orientations, the situation becomes different. Table 6 and 7 show the eigenvalue sensitivity for the laminates, $[\pm 45^\circ]_{2S}$ and $[0^\circ]_{8T}$. As the lamination angles are tailored for specific stiffness, we can see major influence of design variables other than E_I on the eigenvalues in the Tables below. Again, sensitivities whose magnitudes are over 0.1 are shaded in the tables.

Table 6. Eigenvalue sensitivities, $\partial \bar{\lambda}_j / \partial \bar{\theta}$ for laminate, $[\pm 45^\circ]_{2S}$

	1 st mode	2 nd mode	3 rd mode	4 th mode	5 th mode
E_I	0.5050	0.6579	0.4767	0.5233	0.6485
E_2	0.0534	0.0532	0.0415	0.0482	0.0667
ν_{12}	0.0144	-0.0089	0.0063	0.0065	-0.0036
G_{I2}	0.2130	0.0348	0.2530	0.1947	0.0378

Table 7. Eigenvalue sensitivities, $\partial \bar{\lambda}_j / \partial \bar{\theta}$ for laminate, $[0^\circ]_{8T}$

	1 st mode	2 nd mode	3 rd mode	4 th mode	5 th mode
E_I	0.7128	0.3891	0.0870	0.7076	0.6047
E_2	0.0013	0.0158	0.8721	0.0015	0.0184
ν_{12}	0.0033	0.0026	0.0031	0.0034	0.0022
G_{I2}	0.0010	0.3702	0.2232	0.0069	0.1172

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DESCRIPTION OF COMPUTATIONAL ROUTINES

All the procedures in this study are written in MATLAB. Brief descriptions of the major steps and variables are given in this chapter and source listing is attached in Appendix.

A. Main Routine

The main routine (*main_procedure_for_optim.m*) defines the starting vector of the normalized design variables (n_DV) and information of the specimen: specimen length in x-direction (La) and in y-direction (Lb), number of finite element elements in x-direction (Na) and in y-direction (Nb), number of layer in the laminate (Nl), lamination angles (Ang), density of material (ρ), and unit thickness of lamina (t_{layer}). Using the information of the specimen, the main routine calls ‘*modeling_complate.m*’ to generate information for finite elements such as the coordinate of the grid points and the element connection of the finite elements, etc.

Experimental results such as natural frequencies and mode shapes are to be imported. As explained before, reference analytical data are prepared to substitute temporarily for the experimental natural frequencies and mode shapes. Upper and lower bounds for design variables (ub and lb) and several option parameters ($TolFun$, $TolCon$, and etc.) are specified. Constrained minimization routine, *fmincom.m*, is called. As input parameters for this routine, two additional routines are prepared. One is for the performance index calculation (*pi_fv.m*) and the other is a routine to define nonlinear constraints between design variables (*restr_eng_const.m*).

B. Performamnce index calculation

The routine, *pi_fv.m*, is repeatedly called during the minimization process of the performance index. This routine, *pi_fv.m*, calls the routine, *sol_fv.m*, which returns the

analytical natural frequencies and the mode shapes. The performance index is calculated using the experimental and analytic modal data. The variable, *hist*, keeps the record of progress of optimization such as performance index and design variables.

C. Solve Eigenproblem

The routine, *sol_fv.m*, performs element matrix generation, and assembly to generate global matrices, apply boundary condition, and call eigenproblem solver. To find lower several eigenvalues and mass normalized eigenvectors, the routine, *eigsn.m*, is used. It is a modified routine of *eig.m* and *eigs.m*. Analytical eigenvalues and eigenvectors up to a certain mode (here 5th) are returned to the routine, *pi_fv.m*. From the eigenvector, vertical component at the center node of finite elements are chosen and rearranged to be used for comparison with the experimental mode shapes.

D. Finite element generation

There are several routines for finite element generation as follows:

<i>. MeKe.m</i>	generate element stiffness and mass matrices
<i>. dKedDVi.m</i>	generate derivatives of element stiffness matrix
<i>. D_matrix.m</i>	calculate bending stiffness of laminated plates
<i>. D_sen_matrix_wrt_DVi.m</i>	calculate derivatives of D matrix
<i>. sen_eigvalue.m</i>	calculate sensitivities of eigenvalue
<i>. shape_iso9.m</i>	calculate value and derivatives of the shape function of 9-node isoparametric element

V. CONCLUSION AND RECOMMENDATION

A method of obtaining the mechanical properties of the orthotropic lamina is presented along with computational routines. Differences between the natural frequencies from mathematical model and those from experiment are minimized by updating the four mechanical stiffness of lamina. The frequency difference in each mode is weighted based on the modal assurance criteria. A simple vibration test to obtain the natural frequencies and mode shapes can be a substitute for a series of static coupon test to characterize the four mechanical stiffness constants.

This study utilizes finite element analysis using nine-node Reissner-Mindlin plate element, the classical lamination theory, and the first order shear deformation theory. Each procedure is coded in MATLAB, which is included in this report. The MATLAB built-in function, *fmincon*, is used to minimize the performance index. A numerical example is given to demonstrate the performance and usefulness of this scheme.

It is an inverse problem to find the four design variables which can match the dynamic response of a mathematical model of a specimen with the real experimental one. It is necessary, therefore, to have experimental natural frequencies and mode shapes in addition to the computational tools. Future work to get the experimental data are recommended to complete this work for characterization of the mechanical stiffness constants of orthotropic lamina.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- [1] Stephen W. Tsai, 1966, *Mechanics of Composite Materials, Part II, Theoretical Aspects*, Air Force Wright Laboratory Technical Report AFML-TR-66-149
- [2] R. M. Jones, 1999, *Mechanics of composite materials*, 2nd ed., Taylor and Francis, Inc.
- [3] J. M. Whitney, 1967, "Elastic Moduli of Unidirectional Composites with Anisotropic Filaments," *J. of Composite Materials*, pp. 188-193
- [4] J.L. Wearing and C. Patterson, 1981, "Vibration testing of composite materials," *Proc. 1st Int. Conference of Composite Structures*, pp.463-474
- [5] L.R. Deobald and R.F. Gibson, 1988, "Determination of elastic constants of orthotropic plates by a modal analysis/Rayleigh-Ritz technique, *J. of Sound and Vibration*, Vol. 124, pp.269-283
- [6] T. C. Lai and K. H. Ip, 1996, "Parameter estimation of orthotropic plates by Bayesian sensitivity analysis," *Computational Structures*, vol. 34, 1996, pp.29-42
- [7] Dun-xiang Lin, Rong-gen Ni, and R.D. Adams, 1986, "The finite element technique for predicting the natural frequencies, mode shapes and damping values of filamentary composite plates," *Applied Mathematics and Mechanics*, Vol. 7, pp. 197-213
- [8] W. P. Dewilde and H. Sol, 1987, "Anisotropic material identification using measured resonant frequencies of rectangular composite plates," *Composite Structures*, Vol. 4, pp. 317-324
- [9] P. Peterson, C. M. Mota Soares, Moreira De Freitas, and A. L. Araujo, 1993, "Identification of material properties of composite plate specimens," *Composite Structures*, Vol. 25, pp. 277-285
- [10] R. Setiawan, 2004, *Composite plate mechanical characterization through dynamic tests*, Ph. D. thesis, School of Engineering Sciences, Univ. of Southampton
- [11] R. J. Allemang, and D. L. Brown, 1982, "A correlation coefficient for modal vector analysis," *Proceedings of 1st International Modal Analysis Conference*, pp.110-116.

- [12] O. O. Ochoa and J. N. Reddy, 1992, *Finite element analysis of composite laminates*, Kluwer Academic Publishers
- [13] Y. W. Kwon and H. C. Bang, 2000, *The finite element method using MATLAB*, CRC Press LLC
- [14] K. -J. Bathe, 1982, *Finite element procedures in engineering analysis*, Prentice-Hall, Inc.
- [15] M. I. Friswell and J. E. Mottershead, 1995, *Finite element model updating in structural dynamics*, Kluwer Academic Publishers
- [16] R. Fletcher, 1987, *Practical Methods of Optimization*, John Wiley and Sons
- [17] The MathWorks, Inc., 2007, *Optimization Toolbox User's Guide*, Revised for Version 3.1.2 (Release 2007b), <http://www.mathworks.com/products/optimization>

APPENDIX

A. main_procedure_for_optim

```
%----- main_procedure_for_optim.m -----
%
% main program for mechanical property identification
%
%   for laminate, [+/-45/0/0/90]s
%
% Overall procedure
%
%   0) define the design variables: DV's
%       - with proper normalization
%   1) describe the specimen: run "modeling_complate.m"
%       - composite laminate, grid point definition, element connection,,,
%       - declare global variable, *****
%   2) import experimental results
%       - natural frequencies and mode shapes
%       - declare global variables: f_exp, v_exp
%   3) call optimization routine: call "fmincon.m" and "@pi_fv.m"
%   4) in the performance index function, "pi_fv.m"
%       4.0) declare global variables
%       4.1) element matrices: call "MeKe.m"
%           - mass matrix, stiffness matrix
%       4.2) assemble of element matrices and apply boundary condition
%       4.3) solve eigenproblem: call "eigsn.m"
%       4.4) extract eigenvalue and eigenvector(selected dof)
%       4.5) construct the performance index
%           - compare the difference in experiment and analysis
%           - minimization of difference in eigenvalue and ...
%               angle between eigenvector and mode shape
%   8) postprocess the results
%       - plot, etc.
%
% Input information
%
%   [Laminate configuration]
%       - La: length in x-direction (meter)
%       - Lb: length in y-direction (meter)
%       - Na: # of element in x-direction
%       - Nb: # of element in y-direction
%       - Nl: # of layer in composite laminate
%       - Ang: lamination angle
%       - rho: material density (kg/meter^3)
%       - tlayer: lamina thickness(meter)
%
%   [Finite element model]
%       - [Coord-grid]: grid point coordinates, [x1,y1; x2,y2;;; xn,yn]
%       - [Elem]: element connectivity,
%           [EID_1,9-gid's;EID_2,9-gid's;;; EID_n,9-gid's]
%           e.g.(for EID=A shown below)~ Elem=[A,1,15,17,3,8,16,10,2,9]
%       - [Lam]: laminate information, [EID, layer id, thickness, angle]
%           e.g. [Eid_1, 1_1, t_1, a_1; Eid_1, layer_2, t_2, a_2;;;
%               Eid_2, 1_1, t_1, a_1; Eid_1, layer_2, t_2, a_2;;;;;]
%       - [Bc]: applied boundary condition, fixed @x=0
%           [gid, zeros to constrain as many as dof per grid)~ BC
```

```

%
% [Design variables = initial values of material properties]
%   - [DV]: material properties: E1,E2,v12,G12 (SI unit)
%
% [Experimental data]
%   - [f_exp]: natural frequencies in Hz (1 x nmode)
%   - [v_exp]: mode shape (exp_dof x nmode), exp_dof=Na*Nb @center node
% * each mode shape is in the order as follows:
%   [@center of EID1,@center of EID2,,,,,@center of EID(Na*Nb)]'
%
% [Analytical data]
%   - [f_ana]: eigenvalues in Hz during updating design variables
%   - [v_ana]: eigenvectors with selected dof's (exp_dof x nmode)
%
% [Performance index]
%   - J=sum(MAC(i)*sqrt((f_ana(i)-f_exp(i))^2/f_exp(i)^2))
%   - where, MAC(i)=(v_ana(:,i)'*v_exp(:,i))^2/...
%     ((v_ana(:,i)'*v_ana(:,i))*(v_exp(:,i)'*v_exp(:,i)))
%
% 17 Spetember 2007 (jkr)
%
clear all
% clear
% clear global
%
global hist; % history of optimization, fmincon. accumulated @ pi_fv.m
%
%-----
% define design variables and use as initial values
%-----
% define initial value of design variables, DV
%
global DV_r;
%
DV_r=[122.5e9; 7.929e9; 0.329; 3.585e9]; % [E1 E2 v12 G12]
%
n_DV=0.7*[1;1;1;1];
% n_DV=[1.4;0.7;0.6;1.2]; % arbitrary initial value
% n_DV=1.3*[1;1;1;1];
%
% material property DB
% mid rho E1 E2 v12 G12
% 0 1500 122.5e9 7.929e9 0.329 3.585e9; % t=0.150mm Baseline
% 1 1600 207e9 5e9 0.25 2.6e9 % Gr/Ep (Jones)
% 2 1600 122.4552e9 7.92925e9 0.329 3.5854e9; % t=0.125mm Ref.
% 3 1450 122.4552e9 7.92925e9 0.329 3.5854e9; % t=0.13mm, jkr
% 4 1500 108.4e9 7.703e9 0.3193 2.776e9; % t=0.15mm, danbi
%
%-----
% description of specimen, dimensions, lay-up, and etc.
%-----
%
La=0.15; % length in x-direction (meter)
Lb=0.1; % length in y-direction (meter)
Na=6; % # of element in x-direction
Nb=4; % # of element in y-direction
Nl=10; % # of layers including
Ang=[45 -45 0 0 90 90 0 0 -45 45]; % lamination angle
rho=1500; % density (kg/m^3)
tlayer=0.150e-3; % lamina thickness (meter)
%
%-----

```

```

% finite element model data using 9-node isoparametric element
%-----
%
global Coord_grid Elem Lam Bc; % modeling results
%
[Coord_grid,Elem,Lam,Bc]=modeling_complate(La,Lb,Na,Nb,Nl,Ang,tlayer);
%
% parameters for finite element analysis model
%
nnel=9; % number of nodes per element
ndof=3; % dof's per node, [phil,phi2,w]'
edof=nnel*ndof; % dof's per element
ngrid=(2*Na+1)*(2*Nb+1); % total number of nodes (grid point)
t_dof=ndof*ngrid; % total dof
n_elem=Na*Nb; % # of elements
%
global s_info m_info;
%
s_info=[La,Lb,Na,Nb,Nl,Ang,rho,tlayer]; % specimen information, s_info
m_info=[nnel,ndof,edof,ngrid,t_dof,n_elem]; % model information, m_info
%
%-----
% import experiment results
%-----
%
% As there are no available experimental modal data, fictitious
% experimental data are calculated with the reference properties
%
% - for [+45/-45/0/0/90]s laminate
% in the file: "ref_lam1_AR150.mat"
% data: f_r v_r mshp_gid_r lambda_r Phi_r dn_lamdn_DV_r
%
% where, f_r: reference natural frequency up to 5th mode
% v_r: reference mode shape up to 5th mode
% mshp_gid_r: mode shape for plotting
% lambda_r: reference eigenvalues
% Phi_r: reference eigenvectors
% dn_lamdn_DV_r: eigenvalue sensitivity @ref properties
%
global f_exp v_exp mac; % experimental results, weighting
global nmode;
%
nmode=5; % number of modes being considered
%
load ref_lam1_AR150 f_r v_r mshp_gid_r lambda_r Phi_r dn_lamdn_DV_r;
%
f_exp=f_r; v_exp=v_r;
%
%-----
global f_ana v_ana mshp_gid lambda Phi; % defined in the routine, pi_fv_r2
%
%-----
% for optimization using fmincon
% - performance function, pi_fv.m
% - constraints function, restr_eng_const.m
%
% - n_o: optimized design variables, normalized
% - J_o: minimized performance index
%-----
%
lb=0.5*[1;1;1;1];

```

```

ub=1.5*[1;1;1;1];
%
% my_opt=optimset...
% ('Display','notify','FunValCheck','on','TolFun',1e-8,'TolCon',1e-8);
my_opt=optimset...
% ('Display','iter','FunValCheck','on','TolFun',1e-8,'TolCon',1e-8);
%
dfile=input('diary file name: ', 's');
save(dfile, 'hist') % save raw data in 'dfile.mat'
diary (dfile)
disp('-----')
disp('optimization using [45/-45/0/0/90]s laminate')
disp('-----')
disp(['start design variables = [ ', num2str(n_DV), ' ]'])
disp('-----')
disp(['start natural frequency = [ ', num2str(f_exp), ' ]'])
disp('-----')
[n_o,J_o,exitflag_o,output_o]=...
    fmincon(@pi_fv,n_DV,[],[],[],[],lb,ub,@restr_eng_const,my_opt);
disp('-----')
disp(['optim design variables = [ ', num2str(n_o), ' ]'])
disp('-----')
disp(['optim natural frequency = [ ', num2str(f_ana), ' ]'])
disp('-----')
f_ratio=(f_ana./f_exp)*100;
disp(['optim natural frequency = [ ', num2str(f_ratio), ' ]'])
disp('-----')
disp('                                END                                ')
disp('-----')
diary off
%
%-----
% optimization history plot
%-----
[fid,message]=fopen(dfile,'r'); % open data file
%
% check file open
%
if ~isempty(message);
    disp('error!!! To find/open diary file')
    break
end
%
frewind(fid);
%
% move to location of iteration data
%
while 1
    n_line=fgetl(fid);
    [aaa,line_length]=size(n_line);
    if(line_length < 25);
        n_line=fgetl(fid);
    end
    if n_line(2:25) == 'Iter F-count          f(x)';
        break
    end
end
%
% Initialize storage:
xx_iter=[];
xx_fcount=[];
xx_f=[];

```

```

%
% read iteration starting status
%
n_line=fgetl(fid);
    xx_iter=eval(n_line(1:5));
    xx_fcount=eval(n_line(6:12));
    xx_f=eval(n_line(13:25));
    xx(1,:)=[xx_iter, xx_fcount, xx_f];
%
% read iteration history
%
n_iter=1;
n_line=fgetl(fid);
while (eval(n_line(1:5))==n_iter)
    xx_iter=eval(n_line(1:5));
    xx_fcount=eval(n_line(6:12));
    xx_f=eval(n_line(13:25));
    xx(1+n_iter,:)=[xx_iter, xx_fcount, xx_f];
    n_iter=n_iter+1;
    n_line=fgetl(fid);
    [aaa,ok]=str2num(n_line(1:5));
    if ok==0; break; end;
end
%
fclose(fid);
%
% optimization history, opt_hist
% opt_hist=[iter#, Function_call_count, f_value, DV_1~DV_4, f1~f5]
%
opt_hist=[xx(:,1),xx(:,2),hist(xx(:,2),:)]];
%
figure
plot(opt_hist(:,(4:7)))
xlabel('Iteration Number')
ylabel('Normalized design variables')
title('Progress of Optimization')
legend('E1','E2','v12','G12')
grid on
%
figure
plot(opt_hist(:,3))
xlabel('Iteration Number')
ylabel('Performance Index')
title('Progress of Optimization')
legend('PI')
grid on
%
figure
plot(opt_hist(:,(8:12)))
xlabel('Iteration Number')
ylabel('Natural Frequency(Hz)')
title('Progress of Optimization')
legend('f1','f2','f3','f4','f5')
grid on
%
%----- end of main_procedure_for-optim.m -----
%

```

```
%----- modeling_complate.m -----  
%  
function [Coord_grid,Elem,Lam,Bc]...  
    =modeling_complate(La,Lb,Na,Nb,Nl,Ang,tlayer)  
%  
% Input preparation for composite plate: La x Lb clamped at root (at x=0)  
%  
% [Input parameters]  
% - La: length in x-direction (meter)  
% - Lb: length in y-direction (meter)  
% - Na: # of element in x-direction  
% - Nb: # of element in y-direction  
% - Nl: # of layer in composite laminate  
% - Ang: lamination angle (note: reverse)  
% - rho: material density (kg/meter^3)  
% - tlayer: lamina thickness(meter)  
% - DV: material properties: E1,E2,v12,G12 (SI)  
%  
% [Output parameters]  
% - Coord_grid:(x, y)~ coordinates of grid points in sequential order  
% - Elem:(EID, grid_id's)~ element connectivity  
%   e.g.(for EID=A shown below)~ Elem=[A,1,15,17,3,8,16,10,2,9]  
% - Lam :(EID, layer id, thickness, angle)~ laminate information  
% - Bc : (grid id, zeros to constrain as many as dof per grid)~ BC  
  
y  
^  
|  
|  
7---14---21-----+-----  
| | I | I  
6===C===20===I===I      1,2,3,,, : GID  
| | I | I  
5---12---19----+--+--I----+  
| | I | I  
4===B===18====|====I      A,B,C,,, : EID  
| | I | I  
3---10---17----+--+--I----+  
| | I | I  
2===A===16====|====I  
| | I | I  
1---8---15-----+----->x  
  
% 17 September 2007 (jkr)  
%  
Coord_grid=zeros((2*Na+1)*(2*Nb+1),2);  
for i=1:2*Na+1  
    for j=1:2*Nb+1  
        Coord_grid((i-1)*(2*Nb+1)+j,:)= [La/2/Na*(i-1), Lb/2/Nb*(j-1)];  
    end  
end  
%  
Elem=zeros(Na*Nb,10);  
for i=1:Na  
    for j=1:Nb  
        fst=(j-1)*2+(i-1)*2*(2*Nb+1)+1; % first grid id for each element  
        Eid=(i-1)*Nb+j; % element id
```

```

        Elem(Eid,:)=...
[Eid,... % element id
fst,      fst+2*(2*Nb+1),    fst+2*(2*Nb+2),    fst+2,... % corner gid
fst+(2*Nb+1), fst+2*(2*Nb+1)+1, fst+(2*Nb+1)+2,    fst+1,... % side gid
fst+(2*Nb+1)+1]; % center gid
    end
end
clear fst
%
clear Lam
for i=1:Na
    for j=1:Nb
        L_elm=[(i-1)*Nb+j, Nl, tlayer, Ang(Nl)]; % top layer of an element
        for k=2:Nl; % loop to stack layer
            temp=[(i-1)*Nb+j, (Nl+1)-k, tlayer, Ang((Nl+1)-k)]; % next layer
            Lam((i-1)*Nb+j:(i-1)*Nb+j,:)=Lam; temp];
            L_elm=[L_elm;temp]; % stack layer
        %
            [Lam; (i-1)*Nb+j, (Nl+1)-k, tlayer, Ang((Nl+1)-k)];
        end
        Lam((Nl*Nb*(i-1)+Nl*(j-1)+1):(Nl*Nb*(i-1)+j*Nl),:)=L_elm;
    end
end
%
% Boundary Codition: Clamped along the edge, x=0
%
Bc=zeros(2*Nb+1,4);
for i=1:2*Nb+1
    Bc(i,:)= [i 0 0 0 ];
end
%
clear Eid L_elm temp i j k
%
%-----end of modeling_complate.m -----

```

C. pi_fv

```

%----- pi_fv.m -----
%
function J_all=pi_fv(n_DV)
%
% performance index calculation
%
% 17 September 2007 (jkr)
%
global Coord_grid Elem Lam Bc;      % model data
global nmode;
global f_exp v_exp mac; % experimental results, weighting (from main)
global f_ana v_ana mshp_gid lambda Phi; % to be used in main routine
global DV_r;                % reference value of design variable
global hist;
%
%-----
% obtain eigenvalue, eigenvector, and mode shape
%-----
[f_ana,v_ana,mshp_gid,lambda,Phi]=sol_fv(n_DV);
%
%-----
% performance index construction, J=sum{MACi*del(fi)}
%-----
%
for n=1:nmode
    mac(n,1)=(v_ana(:,n)'*v_exp(:,n))^2/...
        ((v_ana(:,n)'*v_ana(:,n))*(v_exp(:,n)'*v_exp(:,n))); % weighting
    delf(n,1)=sqrt((f_ana(n)-f_exp(n))^2/f_exp(n)^2);          % del freq
end
%
% J_all=ones(1,nmode)*delf; % without mode shape weighting
J_all=mac'*delf;           % with mode shape weighting
%
%-----
% convergence history
% each row for iteration containg, [PI,n_DV,f's]
%-----
%
c_hist=[J_all,n_DV',f_ana'];
hist=[hist;c_hist];
%
%----- end of pi_fv.m -----

```

D. rest_eng_const

```
%----- rest_eng_const.m -----  
%  
function [c,ceq]=restr_eng_const(x)  
%  
global DV_r % baseline engineering constants, [E1 E2 v12 G12]  
%  
% engineering constants:  
%   E1=x(1)*DV_r(1), E2=x(2)*DV_r(2), v12=x(3)*DV_r(3), G12=x(4)*DV_r(4)  
%  
% input x: normalized engineering constants  
%           v12-sqrt(E1/E2) < 0  
%           or (v12)^2-(E1/E2) < 0  
%           or x(2)/x(1)*x(3)^2-DV_r(1)/DV_r(2)/DV_r(3)^2 < 0  
%  
% output c: inequality constraints for fmincon, c(x)<= 0  
%           ceq: equality constraints for fmincon, ceq = 0  
%  
% 17 Spetember 2007 (jkr)  
%  
c=x(2)/x(1)*x(3)^2-DV_r(1)/DV_r(2)/DV_r(3)^2;  
ceq=[];  
%  
%-----end of rest_eng_const.m -----
```

E. sol_fv

```

%----- sol_fv.m -----
%
function [f_ana,v_ana,mshp_gid,lambda,Phi]=sol_fv(n_DV)
%
% obtain eigenvalue and eigenvector for given n_DV
%
% 17 September 2007 (jkr)
%
global DV_r; % reference mechanical properties
global Coord_grid Elem Lam Bc; % model data
global s_info m_info;
global nmode;
global f_exp v_exp mac; % experimental results, weighting
%
La=s_info(1); Lb=s_info(2); % plate size
Na=s_info(3); Nb=s_info(4); % # of elements in each direction, x & y
Nl=s_info(5); Ang=s_info(6:5+Nl); % # of layer and lamination angle
rho=s_info(6+Nl); tlayer=s_info(7+Nl); % density and lamina thickness
%
nnel=m_info(1); % number of nodes per element
ndof=m_info(2); % degrees of freedom per node
edof=m_info(3); % degrees of freedom per element
ngrid=m_info(4); % total number of grids (nodes)
t_dof=m_info(5); % total dof
n_elem=m_info(6); % # of elements
%
DV=n_DV.*DV_r; % convert normalized values to physical ones
%
%-----
% finite element matrices construction
%-----
% assemble element matrix
%
M=zeros(t_dof,t_dof); % initialization of M-matrix
K=zeros(t_dof,t_dof); % initialization of K-matrix
%
for Eid=1:n_elem;
%
[Me,Ke]=MeKe(Eid,DV);
%
gids=Elem(Eid,2:10); % grid id's of the element
idx=feeldof(gids,nnel,ndof); % system dof's of the element
%
M=feasmbll(M,Me,idx); % assemble of system mass matrix
K=feasmbll(K,Ke,idx); % assemble of system stiffness matrix
%
end
%
% apply boundary condition (cantilever plate - LHS clamped)
%
[nr,nc]=size(Bc);
nbc=nr*(nc-1); % total # of dof constrained
Ma=M(nbc+1:t_dof,nbc+1:t_dof); % partitioned mass matrix(BC's applied)
Ka=K(nbc+1:t_dof,nbc+1:t_dof); % partitioned stiffness matrix(BC's applied)
%
clear Eid gids nr nc
%
%-----

```

```

% eigenvalue analysis
%-----
%
[Phi_,D_]=eigsn(sparse(Ka),sparse(Ma),nmode);% few modes with normalization
lambda_=diag(D_);
for ii=1:nmode;      % re-order increasing eigenvalue
lambda(ii,1)=lambda_(nmode-ii+1);
Phi(:,ii)=Phi_(:,nmode-ii+1);
end
%-----
%
% % Alternative routine for solving eigen problem above
% [lambda_,Phi_,Psi]=eign(Ka,Ma);
% lambda=lambda_(1:nmode);
% Phi=Phi_(:,1:nmode);
%
%-----
% rearrange eigenvector to plot the mode shapes
%-----
%
mshp_gid=zeros(ngrid,nmode);      % initialize
v_ana=zeros(n_elem,nmode);        % initialize
%
eigmtx_=Phi(:,1:nmode);            % eigenvectors up to 'nmode' modes
eigmtx=[zeros(nbc,nmode);eigmtx_]; % add zeros at fixed boundary dof's
%
% construct mode shape with w's ( get rid of rotation dof's )
%
% mode shape along first along y then increase x ( along gid ), mshp_gid
%
for i=1:ngrid
    mshp_gid(i,:)=eigmtx(ndof*i,:);    %% for mode shape plot
end
%
%-----
% arrange analysis results to compare with experiments
%-----
%
f_ana=sqrt(lambda(1:nmode))/2/pi;      %% natural frequencies from analysis
%
% eigen vectors @ center of element: nele=(Na)x(Nb)
%
for i=1:Na
for j=1:Nb
    v_ana(j+Nb*(i-1),:)=mshp_gid(2*j+(2*Nb+1)*(2*i-1),:); %% mode shapes
end
end
%
%-----end of sol_fv.m -----

```

F. eigsn

```
%----- eigsn.m -----  
%  
function [V,D]=eigsn(K,M,nmode,tol)  
%  
% [V,D] = eigsn(K, M, nmode)  
%     finds lowest eigenvalues and eigenvectors with mass normalization  
%     nomde : # of eigenvalues to be found  
%     defalut tolerance = 1e-10  
% [V,D] = eigsn(K, M, nmode, tol)  
%     tol : user defined tolerance  
%  
% 17 September 2007(jkr)  
%  
if (nargin == 3)  
    tol = 1e-10;  
end  
%  
% EIGS(A,K,SIGMA,OPTS) and EIGS(A,B,K,SIGMA,OPTS) specify options:  
% OPTS.disp: diagnostic information display level [0 | {1} | 2]  
% option_eigs = optimset('OPTS.disp',2);  
%  
OPTS.disp=0;  
[V,D]=eigs(K,M,nmode,0,OPTS);  
%  
% biorthogonality condition  
%  
for i=1:nmode;  
    V(:,i) = V(:,i)/sqrt(V(:,i)'*M*V(:,i));  
end  
%  
%----- end of eigsn.m -----
```

G. MeKe

```

%----- MeKe.m -----
%
function [Me,Ke]=MeKe(Eid,DV)
%
% Generate element mass matrix(Me)and stiffness matrix(Ke)
% based on the 9-node plate element and corresponding numbering
%
% Dof's are {U}'={phix,phiy,w}'
%
% [INPUT]
% - Eid,Lam,Nl,Elem,Coord_grid,DV,tlayer,rho
% - m_info=[nnel,ndof,edof,ngrid,t_dof,n_elem]; % model information
%
% [OUTPUT]
% - Me: 27x27
% - Ke: 27x27
%
% [variables]
% - B: strain vs. nodal displacement relation matrix
% {ex ey rxy}'=z*Bi*{U}'
% - D: moment vs. curvature relation matrix (cf. Stress-Strain)
% {Mx My Mxy}'=[D]*{kx ky ky}'
% (laminate bending stiffness matrix; D11,D12,D16,,,,D66)
% - Qb: stress vs. strain relation matrix
%
% 17 September 2007 (jkr)
%
global Coord_grid Elem Lam Bc; % model data
global s_info m_info; % specimen and model information
%
La=s_info(1); Lb=s_info(2); % plate size
Na=s_info(3); Nb=s_info(4); % # of elements in each direction, x & y
Nl=s_info(5); Ang=s_info(6:5+Nl); % # of layer and lamination angle
rho=s_info(6+Nl); tlayer=s_info(7+Nl); % density and lamina thickness
%
nnel=m_info(1); % number of nodes per element
ndof=m_info(2); % degrees of freedom per node
edof=m_info(3); % degrees of freedom per element
ngrid=m_info(4); % total number of grids (nodes)
t_dof=m_info(5); % total dof
n_elem=m_info(6); % # of elements
%
Ke=zeros(edof,edof); % initialization of stiffness matrix
Kb=zeros(edof,edof); % init of stiffness matrix (bending)
Ks=zeros(edof,edof); % init of stiffness matrix (transverse shear)
Me=zeros(edof,edof); % initialization of mass matrix
%
[Db,Ds]=D_matrix(Eid,DV); % Moment-Curvature relation matrix, D
%
Gid=Elem(Eid,2:10); % grid ID's for element
xcoord=Coord_grid(Gid,1); ycoord=Coord_grid(Gid,2); % grid coord's
%
%-----
% numerical integration bending stiffness and mass ( 3x3 integration )
%-----
%
clear r wtr s wts
nglxb=3; nglyb=3; % use 3x3 integration rule

```

```

[pt3,wt3]=glq2d2(nglxb,nglyb);      % sampling pts & wts in 2-D
%
for intx=1:nglxb
    r=pt3(intx,1);                    % sampling point in x-axis
    wtr=wt3(intx,1);                  % weight in x-axis
    for inty=1:nglyb
        s=pt3(inty,2);                % sampling point in y-axis
        wts=wt3(inty,2);              % weight in y-axis
    %
    % for each integration points
    %
    % compute shape functions and derivatives at integration point
    [h9,dh9dr,dh9ds]=shape_iso9(r,s);
    %
    % initialize dh/dx and dh/dy
    dh9dx=zeros(1,nnel);
    dh9dy=zeros(1,nnel);
    %
    jacob2=zeros(2,2);                % compute Jacobian
    for ii=1:nnel;
        jacob2(1,1)=jacob2(1,1)+dh9dr(ii)*xcoord(ii);
        jacob2(1,2)=jacob2(1,2)+dh9dr(ii)*ycoord(ii);
        jacob2(2,1)=jacob2(2,1)+dh9ds(ii)*xcoord(ii);
        jacob2(2,2)=jacob2(2,2)+dh9ds(ii)*ycoord(ii);
    end
    %
    detjacob=det(jacob2);              % determinant of Jacobian
    invjacob=inv(jacob2);              % inverse of Jacobian matrix
    %
    for ii=1:nnel;                    % derivatives w.r.t physical coordinate
        dh9dx(ii)=invjacob(1,1)*dh9dr(ii)+invjacob(1,2)*dh9ds(ii);
        dh9dy(ii)=invjacob(2,1)*dh9dr(ii)+invjacob(2,2)*dh9ds(ii);
    end
    %
    % for bending stiffness matrix, Kb
    %
    % Strain-Nodal displacement relation matrix, Bi
    % [ex ey gxy]' = z*[Bi]*[phi_1 phi_2 w]'
    % [Bi]=|          dh9dx    0    0          |
    %       | ...    0    dh9dy    0    ... |
    %       |          dh9dy dh9dx    0    |
    %
    Bi=zeros(3,27);
    for ii=1:nnel
        Bi(1,3*(ii-1)+1)=dh9dx(ii);
        %
        Bi(2,3*(ii-1)+2)=dh9dy(ii);
        %
        Bi(3,3*(ii-1)+1)=Bi(2,3*(ii-1)+2);
        Bi(3,3*(ii-1)+2)=Bi(1,3*(ii-1)+1);
    end
    %
    Kb=Kb+Bi'*Db*Bi*wtr*wts*detjacob; % integration for stiffness matrix
    %
    % for mass matrix
    %
    % Displacement interpolation, N
    % [w]=[N][dof]=[N][... phi_1 phi_2 w ...]'
    %      = [... -zhi -zhi hi ...][... phi_1 phi_2 w ...]'
    %
    % [N]=[-z -z 1]|h1 0 0      hi 0 0      h9 0 0|=[z3][H3]
    %           | 0 h1 0 ... 0 hi 0 ... 0 h9 0|

```

```

%          | 0  0 h1      0  0 hi      0  0 h9|
%
%   int(-t/2~t/2){[z3]}'*[z3]} = (t^3/12)| 1   1   0   | = z33
%          | 1   1   0   |
%          | 0   0 12/t^2|
%
H3=zeros(3,27);
for i=1:nnel
    H3(1,3*(i-1)+1)=h9(i);
    H3(2,3*(i-1)+2)=h9(i);
    H3(3,3*(i-1)+3)=h9(i);
end
tt=tlayer*Nl; % laminate thickness
z33=tt^3/12*[ 1   1   0   ; %
              1   1   0   ;
              0   0 12/tt^2];
%
Me=Me+rho*H3'*z33*H3*wtr*wts*detjacob; % integration for mass matrix
%
end % for loop for integration in y
end % for loop for integration in x
%
Kb=(Kb+Kb')/2; % for symmetry
Me=(Me+Me')/2; % for symmetry
%
%-----
% numerical integration trnasverse shear stiffness ( 2x2 integration )
%-----
%
clear r wtr s wts
nglxs=2; nglys=2; % use 3x3 integration rule
[pt2,wt2]=glqd2(nglxs,nglys); % sampling pts & wts in 2-D
%
for intx=1:nglxs
    r=pt2(intx,1); % sampling point in x-axis
    wtr=wt2(intx,1); % weight in x-axis
    for inty=1:nglys
        s=pt2(inty,2); % sampling point in y-axis
        wts=wt2(inty,2); % weight in y-axis
    end
end
% for each integration points
%
% compute shape functions and derivatives at integration point
[h9,dh9dr,dh9ds]=shape_iso9(r,s);
%
% initialize dh/dx and dh/dy
dh9dx=zeros(1,nnel);
dh9dy=zeros(1,nnel);
%
jacob2=zeros(2,2); % compute Jacobian
for ii=1:nnel;
    jacob2(1,1)=jacob2(1,1)+dh9dr(ii)*xcoord(ii);
    jacob2(1,2)=jacob2(1,2)+dh9dr(ii)*ycoord(ii);
    jacob2(2,1)=jacob2(2,1)+dh9ds(ii)*xcoord(ii);
    jacob2(2,2)=jacob2(2,2)+dh9ds(ii)*ycoord(ii);
end
%
detjacob=det(jacob2); % determinant of Jacobian
invjacob=inv(jacob2); % inverse of Jacobian matrix
%
for ii=1:nnel; % derivatives w.r.t physical coordinate
    dh9dx(ii)=invjacob(1,1)*dh9dr(ii)+invjacob(1,2)*dh9ds(ii);

```

```

        dh9dy(ii)=invjacob(2,1)*dh9dr(ii)+invjacob(2,2)*dh9ds(ii);
    end
    %
    % for shear stiffness matrix, Ks
    %
    % Strain-Nodal displacement relation matrix, Bi
    % [gxz gyz]'= [Bs]*[phi_1 phi_2 w]'
    % [Bs]=| ... -hi 0 dh9dx ... |
    %       | ... 0 -hi dh9dy ... |
    %
    Bs=zeros(2,27);
    for ii=1:nnel
        Bs(1,3*(ii-1)+1)=-h9(ii);
        Bs(1,3*(ii-1)+3)=dh9dx(ii);
        %
        Bs(2,3*(ii-1)+2)=-h9(ii);
        Bs(2,3*(ii-1)+3)=dh9dy(ii);
    end
    %
    Ks=Ks+Bs'*Ds*Bs*wtr*wts*detjacob; % integration for stiffness matrix
    %
    end % for loop for integration in y
    end % for loop for integration in x
    %
    Ks=(Ks+Ks')/2; % for symmetry
    %
    %-----
    % total stiffness matrix with shear correction factor,5/6
    %-----
    Ke=Kb+(5/6)*Ks;
    %
    %----- end of MeKe.m -----

```

H. dKedDV_i: $\frac{\partial K_e}{\partial(DV_i)} = \frac{\partial K_e}{\partial \theta_i}$

```
%----- dKedDVi.m -----
%
function [dKedDV1,dKedDV2,dKedDV3,dKedDV4]=dKedDVi(Eid,DV)
%
% Generate derivatives of element stiffness matrix(Ke) w.r.t DVi
% based on the 9-node plate element and corresponding numbering
%
% Dof's are {U}'={phix,phiy,w}'
%
% [INPUT]
% - Eid,Lam,Nl,Elem,Coord_grid,DV,tlayer,rho
% - m_info=[nnel,ndof,edof,ngrid,t_dof,n_elem]; % model information
%
% [OUTPUT]
% - dKedDV1: 27x27 derivative of stiffness matrix wrt DV1
% - dKedDV2: 27x27 derivative of stiffness matrix wrt DV2
% - dKedDV3: 27x27 derivative of stiffness matrix wrt DV3
% - dKedDV4: 27x27 derivative of stiffness matrix wrt DV4
%
% [variables]
% - B: strain vs. nodal displacement relation matrix
%   {ex ey rxy}'=z*Bi*{U}'
% - D: moment vs. curvature relation matrix (cf. Stress-Strain)
%   {Mx My Mxy}'=[D]*{kx ky ky}'
%   (laminate bending stiffness matrix; D11,D12,D16,,,,D66)
% - Qb: stress vs. strain relation matrix
%
% 2 October 2007 (jkr)
%
global Coord_grid Elem Lam Bc; % model data
global s_info m_info; % specimen and model information
%
La=s_info(1); Lb=s_info(2); % plate size
Na=s_info(3); Nb=s_info(4); % # of elements in each direction, x & y
Nl=s_info(5); Ang=s_info(6:5+Nl); % # of layer and lamination angle
rho=s_info(6+Nl); tlayer=s_info(7+Nl); % density and lamina thickness
%
nnel=m_info(1); % number of nodes per element
ndof=m_info(2); % degrees of freedom per node
edof=m_info(3); % degrees of freedom per element
ngrid=m_info(4); % total number of grids (nodes)
t_dof=m_info(5); % total dof
n_elem=m_info(6); % # of elements
%
dKedDV1=zeros(edof,edof); % initialization of matrix
dKedDV2=zeros(edof,edof); % initialization of matrix
dKedDV3=zeros(edof,edof); % initialization of matrix
dKedDV4=zeros(edof,edof); % initialization of matrix
%
dKbdDV1=zeros(edof,edof); % initialization of matrix
dKbdDV2=zeros(edof,edof); % initialization of matrix
dKbdDV3=zeros(edof,edof); % initialization of matrix
dKbdDV4=zeros(edof,edof); % initialization of matrix
%
dKsdDV1=zeros(edof,edof); % initialization of matrix
dKsdDV2=zeros(edof,edof); % initialization of matrix
```

```

dKsdDV3=zeros(edof,edof);    % initialization of matrix
dKsdDV4=zeros(edof,edof);    % initialization of matrix
%
[dDbd1,dDsd1,dDbd2,dDsd2,dDbd3,dDsd3,dDbd4,dDsd4]=...
    D_sen_matrix_wrt_DVi(Eid,DV); % Derivative of D-matrices
%
Gid=Elem(Eid,2:10);          % grid ID's for element
xcoord=Coord_grid(Gid,1); ycoord=Coord_grid(Gid,2); % grid coord's
%
%-----
% numerical integration of derivatives of bending stiffness ( 3x3 int. )
%-----
%
clear r wtr s wts
nglxb=3; nglyb=3;            % use 3x3 integration rule
[pt3,wt3]=glqd2(nglxb,nglyb); % sampling pts & wts in 2-D
%
for intx=1:nglxb
    r=pt3(intx,1);            % sampling point in x-axis
    wtr=wt3(intx,1);          % weight in x-axis
    for inty=1:nglyb
        s=pt3(inty,2);        % sampling point in y-axis
        wts=wt3(inty,2);      % weight in y-axis
    %
    % for each integration points
    %
    % compute shape functions and derivatives at integration point
    [h9,dh9dr,dh9ds]=shape_iso9(r,s);
    %
    % initialize dh/dx and dh/dy
    dh9dx=zeros(1,nnel);
    dh9dy=zeros(1,nnel);
    %
    jacob2=zeros(2,2);        % compute Jacobian
    for ii=1:nnel;
        jacob2(1,1)=jacob2(1,1)+dh9dr(ii)*xcoord(ii);
        jacob2(1,2)=jacob2(1,2)+dh9dr(ii)*ycoord(ii);
        jacob2(2,1)=jacob2(2,1)+dh9ds(ii)*xcoord(ii);
        jacob2(2,2)=jacob2(2,2)+dh9ds(ii)*ycoord(ii);
    end
    %
    detjacob=det(jacob2);      % determinant of Jacobian
    invjacob=inv(jacob2);      % inverse of Jacobian matrix
    %
    for ii=1:nnel;            % derivatives w.r.t physical coordinate
        dh9dx(ii)=invjacob(1,1)*dh9dr(ii)+invjacob(1,2)*dh9ds(ii);
        dh9dy(ii)=invjacob(2,1)*dh9dr(ii)+invjacob(2,2)*dh9ds(ii);
    end
    %
    % for bending stiffness matrix, Kb
    %
    % Strain-Nodal displacement relation matrix, Bi
    % [ex ey gxy]'= z*[Bi]*[phi_1 phi_2 w]'
    % [Bi]=|      dh9dx    0    0      |
    %      | ...    0    dh9dy    0    ... |
    %      |      dh9dy    dh9dx    0      |
    %
    Bi=zeros(3,27);
    for ii=1:nnel
        Bi(1,3*(ii-1)+1)=dh9dx(ii);
        %
        Bi(2,3*(ii-1)+2)=dh9dy(ii);

```

```

%
Bi(3,3*(ii-1)+1)=Bi(2,3*(ii-1)+2);
Bi(3,3*(ii-1)+2)=Bi(1,3*(ii-1)+1);
end
%
dKbdDV1=dKbdDV1+Bi'*dDbd1*Bi*wtr*wts*detjacob; % integration for matrix
dKbdDV2=dKbdDV2+Bi'*dDbd2*Bi*wtr*wts*detjacob; % integration for matrix
dKbdDV3=dKbdDV3+Bi'*dDbd3*Bi*wtr*wts*detjacob; % integration for matrix
dKbdDV4=dKbdDV4+Bi'*dDbd4*Bi*wtr*wts*detjacob; % integration for matrix
%
end % for loop for integration in y
end % for loop for integration in x
%
dKbdDV1=(dKbdDV1+dKbdDV1')/2; % for symmetry
dKbdDV2=(dKbdDV2+dKbdDV2')/2; % for symmetry
dKbdDV3=(dKbdDV3+dKbdDV3')/2; % for symmetry
dKbdDV4=(dKbdDV4+dKbdDV4')/2; % for symmetry
%
%-----
% num integration of derivatives fo trans shear stiffness ( 2x2 integ )
%-----
%
clear r wtr s wts
nglxs=2; nglys=2; % use 3x3 integration rule
[pt2,wt2]=glqd2(nglxs,nglys); % sampling pts & wts in 2-D
%
for intx=1:nglxs
    r=pt2(intx,1); % sampling point in x-axis
    wtr=wt2(intx,1); % weight in x-axis
    for inty=1:nglys
        s=pt2(inty,2); % sampling point in y-axis
        wts=wt2(inty,2); % weight in y-axis
    %
    % for each integration points
    %
    % compute shape functions and derivatives at integration point
    [h9,dh9dr,dh9ds]=shape_iso9(r,s);
    %
    % initialize dh/dx and dh/dy
    dh9dx=zeros(1,nnel);
    dh9dy=zeros(1,nnel);
    %
    jacob2=zeros(2,2); % compute Jacobian
    for ii=1:nnel;
        jacob2(1,1)=jacob2(1,1)+dh9dr(ii)*xcoord(ii);
        jacob2(1,2)=jacob2(1,2)+dh9dr(ii)*ycoord(ii);
        jacob2(2,1)=jacob2(2,1)+dh9ds(ii)*xcoord(ii);
        jacob2(2,2)=jacob2(2,2)+dh9ds(ii)*ycoord(ii);
    end
    %
    detjacob=det(jacob2); % determinant of Jacobian
    invjacob=inv(jacob2); % inverse of Jacobian matrix
    %
    for ii=1:nnel; % derivatives w.r.t physical coordinate
        dh9dx(ii)=invjacob(1,1)*dh9dr(ii)+invjacob(1,2)*dh9ds(ii);
        dh9dy(ii)=invjacob(2,1)*dh9dr(ii)+invjacob(2,2)*dh9ds(ii);
    end
    %
    % for shear stiffness matrix, Ks
    %
    % Strain-Nodal displacement relation matrix, Bi
    [gxz gyz]=' [Bs]*[phi_1 phi_2 w]'

```

```

% [Bs]=| ... -hi 0 dh9dx ... |
%      | ... 0 -hi dh9dy ... |
%
Bs=zeros(2,27);
for ii=1:nnel
    Bs(1,3*(ii-1)+1)=-h9(ii);
    Bs(1,3*(ii-1)+3)=dh9dx(ii);
    %
    Bs(2,3*(ii-1)+2)=-h9(ii);
    Bs(2,3*(ii-1)+3)=dh9dy(ii);
end
%
dKsdDV1=dKsdDV1+B's'*dDsd1*B's*wtr*wts*detjacob; % integration for matrix
dKsdDV2=dKsdDV2+B's'*dDsd2*B's*wtr*wts*detjacob; % integration for matrix
dKsdDV3=dKsdDV3+B's'*dDsd3*B's*wtr*wts*detjacob; % integration for matrix
dKsdDV4=dKsdDV4+B's'*dDsd4*B's*wtr*wts*detjacob; % integration for matrix
%
end % for loop for integration in y
end % for loop for integration in x
%
dKsdDV1=(dKsdDV1+dKsdDV1')/2; % for symmetry
dKsdDV2=(dKsdDV2+dKsdDV2')/2; % for symmetry
dKsdDV3=(dKsdDV3+dKsdDV3')/2; % for symmetry
dKsdDV4=(dKsdDV4+dKsdDV4')/2; % for symmetry
%
%-----
% total stiffness matrix with shear correction factor,5/6
%-----
dKedDV1=dKbdDV1+(5/6)*dKsdDV1;
dKedDV2=dKbdDV2+(5/6)*dKsdDV2;
dKedDV3=dKbdDV3+(5/6)*dKsdDV3;
dKedDV4=dKbdDV4+(5/6)*dKsdDV4;
%
%----- end of dKedDV1.m -----

```

I. D_matrix

```
%----- D_matrix.m -----
%
function [Db,Ds]=D_matrix(Eid,DV)
%
% D_matrices of each element
%
% [INPUT]
%   - Eid, DV
%
% [OUTPUT]
%   - Db = | D11 D12 D16 | "bending stiffness"
%           | D12 D22 D26 |
%           | D16 D26 D66 |
%   - Ds = | D55 D54 | "transverse shear stiffness"
%           | D54 D44 |
%
% 17 September 2007 (jkr)
%
global Coord_grid Elem Lam Bc;      % model data
global s_info m_info;              % specimen and model information
%
Nl=s_info(5);
%
Lam=Lam(Nl*(Eid-1)+1:Nl*Eid,:,:,:);
layer_id=Lam_(1:Nl,2);
layer_t=Lam_(1:Nl,3);
layer_angle=Lam_(1:Nl,4);
%
E1=DV(1);
E2=DV(2);
v12=DV(3);
G12=DV(4);
v21=v12*E2/E1;
%
tt=sum(layer_t);    % laminate thickness
%
% in-plane property
%
Q11=E1/(1-v12*v21);
Q22=E2/(1-v12*v21);
Q12=v12*E2/(1-v12*v21);
Q66=G12;
%
% trnasverse shear properties are expressed in Q66 and 0.5*(Q11-Q12)
%   (assumed transversely isotropic)
%
z(1)=-tt/2;
zcoord=z(1);
for ii=1:Nl
    zcoord=zcoord+layer_t(ii);
    z(ii+1)=zcoord;
end
%
Db=zeros(3,3);
Ds=zeros(2,2);
%B=zeros(3,3);
%A=zeros(3,3);
for ii=1:Nl
```

```

theta=layer_angle(ii)*pi/180;
cs=cos(theta); sn=sin(theta);
%
Qb11=Q11*cs^4+2*(Q12+2*Q66)*sn^2*cs^2+Q22*sn^4;
Qb12=(Q11+Q22-4*Q66)*sn^2*cs^2+Q12*(sn^4+cs^4);
Qb22=Q11*sn^4+2*(Q12+2*Q66)*sn^2*cs^2+Q22*cs^4;
Qb16=(Q11-Q12-2*Q66)*sn*cs^3+(Q12-Q22+2*Q66)*sn^3*cs;
Qb26=(Q11-Q12-2*Q66)*sn^3*cs+(Q12-Q22+2*Q66)*sn*cs^3;
Qb66=(Q11+Q22-2*Q12-2*Q66)*sn^2*cs^2+Q66*(sn^4+cs^4);
%
Qs55=Q66*cs^2+0.5*(Q11-Q12)*sn^2;
Qs54=-Q66*cs*sn+0.5*(Q11-Q12)*cs*sn;
Qs44=Q66*sn^2+0.5*(Q11-Q12)*cs^2;
%
Qb=[Qb11 Qb12 Qb16;
     Qb12 Qb22 Qb26;
     Qb16 Qb26 Qb66];
%
Qs=[Qs55 Qs54;
     Qs54 Qs44];
%
Db=Db+1/3*Qb*(z(ii+1)^3-z(ii)^3);
Ds=Ds+Qs*(z(ii+1)-z(ii));
%
B=B+1/2*Qb*(z(ii+1)^2-z(ii)^2);
%
A=A+Qb*(z(ii+1)-z(ii));
end
%----- end of D_matrix.m -----

```

J. D_sen_matrix_wrt_DVi

```
%----- D_sen_matrix_wrt_DVi.m -----
%
function [dDbd1,dDsd1,dDbd2,dDsd2,dDbd3,dDsd3,dDbd4,dDsd4]=...
    D_sen_matrix_wrt_DVi(Eid,DV)
%
% Sensitivity of D_matrices(Db and Ds) w.r.t. DV1=E1,DV2=E2,DV3=v12,DV3=G12
%
% [INPUT]
%     - Eid, Nl, Lam, DV
%
% [OUTPUT]
%     - dDbdi =      d      | D11 D12 D16 |
%               ----- | D12 D22 D26 |
%               d(DVi) | D16 D26 D66 |
%
%     - dDsdi =      d      | D55  D54 |
%               ----- |          |
%               d(DVi) | D54  D44 |
%
% 1 October 2007 (jkr)
%
global Coord_grid Elem Lam Bc;      % model data
global s_info m_info
%
La=s_info(1); Lb=s_info(2); % plate size
Na=s_info(3); Nb=s_info(4); % # of elements in each direction, x & y
Nl=s_info(5); Ang=s_info(6:5+Nl); % # of layer and lamination angle
rho=s_info(6+Nl); tlayer=s_info(7+Nl); % density and lamina thickness
%
nnel=m_info(1);      % number of nodes per element
ndof=m_info(2);      % degrees of freedom per node
edof=m_info(3);      % degrees of freedom per element
ngrid=m_info(4);     % total number of grids (nodes)
t_dof=m_info(5);     % total dof
n_elem=m_info(6);    % # of elements
%
Lam=Lam(Nl*(Eid-1)+1:Nl*Eid,:,:,:);
layer_id=Lam_(1:Nl,2);
layer_t=Lam_(1:Nl,3);
layer_angle=Lam_(1:Nl,4);
%
E1=DV(1);
E2=DV(2);
v12=DV(3);
G12=DV(4);
v21=v12*E2/E1;
%
tt=sum(layer_t);      % laminate thickness
%
% on-axis properties
%
%   in-plane
%       Q11=E1/(1-v12*v21);
%       Q22=E2/(1-v12*v21);
%       Q12=v12*E2/(1-v12*v21);
%       Q66=G12;
%
%   trnasverse shear (assumed transversely isotropic)
```

```

%      Q55=Q66
%      Q44=0.5*(Q11-Q12)

%
% Derivatives wrt DV1; d(Q11)/d(E1), d(Q22)/d(E1),,,
%
dQ11d1=1/(1-v12*v21)-v12*v21/(1-v12*v21)^2;
dQ22d1=-v21^2/(1-v12*v21)^2;
dQ12d1=-v12*v21^2/(1-v12*v21)^2;
dQ66d1=0;
%
% Derivatives wrt DV2; d(Q11)/d(E2), d(Q22)/d(E2),,,
%
dQ11d2=v12^2/(1-v12*v21)^2;
dQ22d2=1/(1-v12*v21)+v12*v21/(1-v12*v21)^2;
dQ12d2=v12/(1-v12*v21)+v12^2*v21/(1-v12*v21)^2;
dQ66d2=0;
%
% Derivatives wrt DV3; d(Q11)/d(v12), d(Q22)/d(v12),,,
%
dQ11d3=2*v12*E2/(1-v12*v21)^2;
dQ22d3=2*v21*E2/(1-v12*v21)^2;
dQ12d3=E2/(1-v12*v21)+2*v12*v21*E2/(1-v12*v21)^2;
dQ66d3=0;
%
% Derivatives wrt DV1; d(Q11)/d(E2), d(Q22)/d(E2),,,
%
dQ11d4=0;
dQ22d4=0;
dQ12d4=0;
dQ66d4=1;
%
z(1)=-tt/2;
zcoord=z(1);
for ii=1:Nl
    zcoord=zcoord+layer_t(ii);
    z(ii+1)=zcoord;
end
%
dDbd1=zeros(3,3);    dDsd1=zeros(2,2);    % initialize
dDbd2=zeros(3,3);    dDsd2=zeros(2,2);    % initialize
dDbd3=zeros(3,3);    dDsd3=zeros(2,2);    % initialize
dDbd4=zeros(3,3);    dDsd4=zeros(2,2);    % initialize
%-----
% sensitivity wrt DV1
%-----
for ii=1:Nl
    theta=layer_angle(ii)*pi/180;
    cs=cos(theta); sn=sin(theta);
%
dQb11d1=dQ11d1*cs^4+2*(dQ12d1+2*dQ66d1)*sn^2*cs^2+dQ22d1*sn^4;
dQb12d1=(dQ11d1+dQ22d1-4*dQ66d1)*sn^2*cs^2+dQ12d1*(sn^4+cs^4);
dQb22d1=dQ11d1*sn^4+2*(dQ12d1+2*dQ66d1)*sn^2*cs^2+dQ22d1*cs^4;
dQb16d1=(dQ11d1-dQ12d1-2*dQ66d1)*sn*cs^3+(dQ12d1-dQ22d1+2*dQ66d1)*sn^3*cs;
dQb26d1=(dQ11d1-dQ12d1-2*dQ66d1)*sn^3*cs+(dQ12d1-dQ22d1+2*dQ66d1)*sn*cs^3;
dQb66d1=(dQ11d1+dQ22d1-2*dQ12d1-2*dQ66d1)*sn^2*cs^2+dQ66d1*(sn^4+cs^4);
%
dQs55d1=dQ66d1*cs^2+0.5*(dQ11d1-dQ12d1)*sn^2;
dQs54d1=-dQ66d1*cs*sn+0.5*(dQ11d1-dQ12d1)*cs*sn;
dQs44d1=dQ66d1*sn^2+0.5*(dQ11d1-dQ12d1)*cs^2;
%
dQbd1=[dQb11d1 dQb12d1 dQb16d1;

```

```

        dQb12d1 dQb22d1 dQb26d1;
        dQb16d1 dQb26d1 dQb66d1];
%
    dQsd1=[dQs55d1 dQs54d1;
           dQs54d1 dQs44d1];
%
    dDbd1=dDbd1+1/3*dQbd1*(z(ii+1)^3-z(ii)^3);
    dDsd1=dDsd1+dQsd1*(z(ii+1)-z(ii));
end
%
%-----
% sensitivity wrt DV2
%-----
for ii=1:Nl
    theta=layer_angle(ii)*pi/180;
    cs=cos(theta); sn=sin(theta);
%
    dQb11d2=dQ11d2*cs^4+2*(dQ12d2+2*dQ66d2)*sn^2*cs^2+dQ22d2*sn^4;
    dQb12d2=(dQ11d2+dQ22d2-4*dQ66d2)*sn^2*cs^2+dQ12d2*(sn^4+cs^4);
    dQb22d2=dQ11d2*sn^4+2*(dQ12d2+2*dQ66d2)*sn^2*cs^2+dQ22d2*cs^4;
    dQb16d2=(dQ11d2-dQ12d2-2*dQ66d2)*sn*cs^3+(dQ12d2-dQ22d2+2*dQ66d2)*sn^3*cs;
    dQb26d2=(dQ11d2-dQ12d2-2*dQ66d2)*sn^3*cs+(dQ12d2-dQ22d2+2*dQ66d2)*sn*cs^3;
    dQb66d2=(dQ11d2+dQ22d2-2*dQ12d2-2*dQ66d2)*sn^2*cs^2+dQ66d2*(sn^4+cs^4);
%
    dQs55d2=dQ66d2*cs^2+0.5*(dQ11d2-dQ12d2)*sn^2;
    dQs54d2=-dQ66d2*cs*sn+0.5*(dQ11d2-dQ12d2)*cs*sn;
    dQs44d2=dQ66d2*sn^2+0.5*(dQ11d2-dQ12d2)*cs^2;
%
    dQbd2=[dQb11d2 dQb12d2 dQb16d2;
           dQb12d2 dQb22d2 dQb26d2;
           dQb16d2 dQb26d2 dQb66d2];
%
    dQsd2=[dQs55d2 dQs54d2;
           dQs54d2 dQs44d2];
%
    dDbd2=dDbd2+1/3*dQbd2*(z(ii+1)^3-z(ii)^3);
    dDsd2=dDsd2+dQsd2*(z(ii+1)-z(ii));
end
%
%-----
% sensitivity wrt DV3
%-----
for ii=1:Nl
    theta=layer_angle(ii)*pi/180;
    cs=cos(theta); sn=sin(theta);
%
    dQb11d3=dQ11d3*cs^4+2*(dQ12d3+2*dQ66d3)*sn^2*cs^2+dQ22d3*sn^4;
    dQb12d3=(dQ11d3+dQ22d3-4*dQ66d3)*sn^2*cs^2+dQ12d3*(sn^4+cs^4);
    dQb22d3=dQ11d3*sn^4+2*(dQ12d3+2*dQ66d3)*sn^2*cs^2+dQ22d3*cs^4;
    dQb16d3=(dQ11d3-dQ12d3-2*dQ66d3)*sn*cs^3+(dQ12d3-dQ22d3+2*dQ66d3)*sn^3*cs;
    dQb26d3=(dQ11d3-dQ12d3-2*dQ66d3)*sn^3*cs+(dQ12d3-dQ22d3+2*dQ66d3)*sn*cs^3;
    dQb66d3=(dQ11d3+dQ22d3-2*dQ12d3-2*dQ66d3)*sn^2*cs^2+dQ66d3*(sn^4+cs^4);
%
    dQs55d3=dQ66d3*cs^2+0.5*(dQ11d3-dQ12d3)*sn^2;
    dQs54d3=-dQ66d3*cs*sn+0.5*(dQ11d3-dQ12d3)*cs*sn;
    dQs44d3=dQ66d3*sn^2+0.5*(dQ11d3-dQ12d3)*cs^2;
%
    dQbd3=[dQb11d3 dQb12d3 dQb16d3;
           dQb12d3 dQb22d3 dQb26d3;
           dQb16d3 dQb26d3 dQb66d3];
%
    dQsd3=[dQs55d3 dQs54d3;

```

```

        dQs54d3 dQs44d3];
%
    dDbd3=dDbd3+1/3*dQbd3*(z(ii+1)^3-z(ii)^3);
    dDsd3=dDsd3+dQsd3*(z(ii+1)-z(ii));
end
%
%-----
% sensitivity wrt DV4
%-----
for ii=1:Nl
    theta=layer_angle(ii)*pi/180;
    cs=cos(theta); sn=sin(theta);
%
    dQb11d4=dQ11d4*cs^4+2*(dQ12d4+2*dQ66d4)*sn^2*cs^2+dQ22d4*sn^4;
    dQb12d4=(dQ11d4+dQ22d4-4*dQ66d4)*sn^2*cs^2+dQ12d4*(sn^4+cs^4);
    dQb22d4=dQ11d4*sn^4+2*(dQ12d4+2*dQ66d4)*sn^2*cs^2+dQ22d4*cs^4;
    dQb16d4=(dQ11d4-dQ12d4-2*dQ66d4)*sn*cs^3+(dQ12d4-dQ22d4+2*dQ66d4)*sn^3*cs;
    dQb26d4=(dQ11d4-dQ12d4-2*dQ66d4)*sn^3*cs+(dQ12d4-dQ22d4+2*dQ66d4)*sn*cs^3;
    dQb66d4=(dQ11d4+dQ22d4-2*dQ12d4-2*dQ66d4)*sn^2*cs^2+dQ66d4*(sn^4+cs^4);
%
    dQs55d4=dQ66d4*cs^2+0.5*(dQ11d4-dQ12d4)*sn^2;
    dQs54d4=-dQ66d4*cs*sn+0.5*(dQ11d4-dQ12d4)*cs*sn;
    dQs44d4=dQ66d4*sn^2+0.5*(dQ11d4-dQ12d4)*cs^2;
%
    dQbd4=[dQb11d4 dQb12d4 dQb16d4;
           dQb12d4 dQb22d4 dQb26d4;
           dQb16d4 dQb26d4 dQb66d4];
%
    dQsd4=[dQs55d4 dQs54d4;
           dQs54d4 dQs44d4];
%
    dDbd4=dDbd4+1/3*dQbd4*(z(ii+1)^3-z(ii)^3);
    dDsd4=dDsd4+dQsd4*(z(ii+1)-z(ii));
end
%----- end of D_sen_matrix_wrt_DVi.m -----

```

K. sen_eigenvalue

```
%----- sen_eigvalue.m -----
%
function dn_lamdn_DV=sen_eigvalue(n_DV)
%
%-----
% sensitivity of eigenvalues w.r.t design variables
%-----
%
global DV_r;
global Coord_grid Elem Lam Bc;          % model data
global s_info m_info;                   % specimen and model information
global nmode;
global f_exp v_exp mac;                  % experimental results, weighting
global f_ana v_ana mshp_gid lambda Phi; % defined in the routine, pi_fv_r2
%
    La=s_info(1); Lb=s_info(2); % plate size
    Na=s_info(3); Nb=s_info(4); % # of elements in each direction, x & y
    Nl=s_info(5); Ang=s_info(6:5+Nl); % # of layer and lamination angle
    rho=s_info(6+Nl); tlayer=s_info(7+Nl); % density and lamina thickness
%
    nnel=m_info(1); % number of nodes per element
    ndof=m_info(2); % degrees of freedom per node
    edof=m_info(3); % degrees of freedom per element
    ngrid=m_info(4); % total number of grids (nodes)
    t_dof=m_info(5); % total dof
    n_elem=m_info(6); % # of elements
%
DV=n_DV.*DV_r;
%
% assemble of stiffness sensitivity matrix
%
dKdDV1=zeros(t_dof,t_dof); % initialization of stiffness derivative wrt DV1
dKdDV2=zeros(t_dof,t_dof); % initialization of stiffness derivative wrt DV2
dKdDV3=zeros(t_dof,t_dof); % initialization of stiffness derivative wrt DV3
dKdDV4=zeros(t_dof,t_dof); % initialization of stiffness derivative wrt DV4
%
for Eid=1:n_elem;
    %
    [dKedDV1,dKedDV2,dKedDV3,dKedDV4]=dKedDVi(Eid,DV);
    %
    gids=Elem(Eid,2:10); % grid id's of the element
    idx=feeldof(gids,nnel,ndof); % system dof's of the element
    %
    dKdDV1=feasmb11(dKdDV1,dKedDV1,idx); % assemble of matrix for DV1
    dKdDV2=feasmb11(dKdDV2,dKedDV2,idx); % assemble of matrix for DV2
    dKdDV3=feasmb11(dKdDV3,dKedDV3,idx); % assemble of matrix for DV3
    dKdDV4=feasmb11(dKdDV4,dKedDV4,idx); % assemble of matrix for DV4
    %
end
%
% apply boundary condition (cantilever plate - LHS clamped)
%
[nr,nc]=size(Bc);
nbc=nr*(nc-1); % total # of dof constrained
dKadDV1=dKdDV1(nbc+1:t_dof,nbc+1:t_dof); % matrix partition (apply BC's)
dKadDV2=dKdDV2(nbc+1:t_dof,nbc+1:t_dof); % matrix partition (apply BC's)
dKadDV3=dKdDV3(nbc+1:t_dof,nbc+1:t_dof); % matrix partition (apply BC's)
dKadDV4=dKdDV4(nbc+1:t_dof,nbc+1:t_dof); % matrix partition (apply BC's)
```

```

%
% normalized sensitivity of eigenvalue for each mode
%
dn_lamdn_DV=zeros(length(n_DV),nmode);
%
for m=1:nmode
    dn_lamdn_DV(1,m)=Phi(:,m)'*dKadDV1*Phi(:,m)*DV_r(1)*1/lambda(m);
    dn_lamdn_DV(2,m)=Phi(:,m)'*dKadDV2*Phi(:,m)*DV_r(2)*1/lambda(m);
    dn_lamdn_DV(3,m)=Phi(:,m)'*dKadDV3*Phi(:,m)*DV_r(3)*1/lambda(m);
    dn_lamdn_DV(4,m)=Phi(:,m)'*dKadDV4*Phi(:,m)*DV_r(4)*1/lambda(m);
end
%
%----- end of sen_eigvalue.m -----

```

L. shape_iso9

```

%----- shape_iso9.m -----
%
function [h9,dh9dr,dh9ds]=shape_iso9(r,s)
%
% Returns the value of following functions at corresponding node(r,s)
%   h9:      shape function of 9-node element
%   dh9dr:   d(h9)/dr
%   dh9ds:   d(h9)/ds
%
% node numbering as shown below:
%
%           ^ s
%           |
%   o-----o-----o
%   | 4       7       | 3
%   |           |
%   o       o       o ---> r
%   | 8       9       | 6
%   |           |
%   o-----o-----o
%   1       5       2
%
% Refer to the Bathe or Kwon and Bang
% No: node number (1 to 9)
% r,s: node coordinates
%
% 15 July 2007 (jkr)
%
h9=zeros(1,9); dh9dr=zeros(1,9); dh9ds=zeros(1,9);
%
% Shape function
%
h9(1)=0.25*(r*r-r)*(s*s-s);
h9(2)=0.25*(r*r+r)*(s*s-s);
h9(3)=0.25*(r*r+r)*(s*s+s);
h9(4)=0.25*(r*r-r)*(s*s+s);
h9(5)=-0.5*(r*r-1)*(s*s-s);
h9(6)=-0.5*(r*r+r)*(s*s-1);
h9(7)=-0.5*(r*r-1)*(s*s+s);
h9(8)=-0.5*(r*r-r)*(s*s-1);
h9(9)=(r*r-1)*(s*s-1);
%
% Derivative of shape function w.r.t "r"
%
dh9dr(1)=0.25*(2*r-1)*(s*s-s);
dh9dr(2)=0.25*(2*r+1)*(s*s-s);
dh9dr(3)=0.25*(2*r+1)*(s*s+s);
dh9dr(4)=0.25*(2*r-1)*(s*s+s);
dh9dr(5)=-0.5*(2*r)*(s*s-s);
dh9dr(6)=-0.5*(2*r+1)*(s*s-1);
dh9dr(7)=-0.5*(2*r)*(s*s+s);
dh9dr(8)=-0.5*(2*r-1)*(s*s-1);
dh9dr(9)=(2*r)*(s*s-1);
%
% Derivative of shape function w.r.t "s"
%
dh9ds(1)=0.25*(r*r-r)*(2*s-1);
dh9ds(2)=0.25*(r*r+r)*(2*s-1);

```

```

dh9ds(3)=0.25*(r*r+r)*(2*s+1);
dh9ds(4)=0.25*(r*r-r)*(2*s+1);
dh9ds(5)=-0.5*(r*r-1)*(2*s-1);
dh9ds(6)=-0.5*(r*r+r)*(2*s);
dh9ds(7)=-0.5*(r*r-1)*(2*s+1);
dh9ds(8)=-0.5*(r*r-r)*(2*s);
dh9ds(9)=(r*r-1)*(2*s);
%
%----- end of shape_iso9.m -----

```

M. **feasmb11**

```
%----- feasmb11.m -----  
%  
function [kk]=feasmb11(kk,k,index)  
%-----  
%   Purpose:  
%     Assembly of element matrices into the system matrix  
%  
%   Synopsis:  
%     [kk]=feasmb11(kk,k,index)  
%  
%   Variable Description:  
%     kk - system matrix  
%     k  - element matrix  
%     index - d.o.f. vector associated with an element  
%-----  
edof = length(index);  
for i=1:edof  
    ii=index(i);  
    for j=1:edof  
        jj=index(j);  
        kk(ii,jj)=kk(ii,jj)+k(i,j);  
    end  
end  
end  
%  
%----- end of feasmb11.m -----
```

N. **feeldof**

```
%----- feeldof.m -----  
%  
function index=feeldof(nd,nnel,ndof)  
%-----  
% Purpose:  
%     Compute system dofs associated with each element  
%  
% Synopsis:  
%     [index]=feeldof(nd,nnel,ndof)  
%  
% Variable Description:  
%     index - system dof vector associated with element "iel"  
%     iel - element number whose system dofs are to be determined  
%     nd - grid numbers of the element  
%     nnel - number of nodes per element  
%     ndof - number of dofs per node  
%-----  
k=0;  
for i=1:nnel  
    start = (nd(i)-1)*ndof;  
    for j=1:ndof  
        k=k+1;  
        index(k)=start+j;  
    end  
end  
%  
%----- end of feeldof.m -----
```

O. feqlqd1

```
%----- feqlqd1.m -----  
%  
function [point1,weight1]=feqlqd1(ngl)  
%-----  
% Purpose:  
%     determine the integration points and weighting coefficients  
%     of Gauss-Legendre quadrature for one-dimensional integration  
%  
% Synopsis:  
%     [point1,weight1]=feqlqd1(ngl)  
%  
% Variable Description:  
%     ngl - number of integration points  
%     point1 - vector containing integration points  
%     weight1 - vector containing weighting coefficients  
%-----  
%  
% initialization  
%  
    point1=zeros(ngl,1);  
    weight1=zeros(ngl,1);  
%  
% find corresponding integration points and weights  
%  
    if ngl==1           % 1-point quadrature rule  
        point1(1)=0.0;  
        weight1(1)=2.0;  
%  
    elseif ngl==2       % 2-point quadrature rule  
        point1(1)=-1/sqrt(3); % point1(1)=-0.577350269189626  
        point1(2)=-point1(1);  
        weight1(1)=1.0;  
        weight1(2)=weight1(1);  
%  
    elseif ngl==3       % 3-point quadrature rule  
        point1(1)=-sqrt(3/5); %point1(1)=-0.774596669241483  
        point1(2)=0.0;  
        point1(3)=-point1(1);  
        weight1(1)=5/9; % weight1(1)=0.555555555555556  
        weight1(2)=8/9; % weight1(1)=0.888888888888889  
        weight1(3)=weight1(1);  
%  
    elseif ngl==4       % 4-point quadrature rule  
        point1(1)=-0.861136311594053;  
        point1(2)=-0.339981043584856;  
        point1(3)=-point1(2);  
        point1(4)=-point1(1);  
        weight1(1)=0.347854845137454;  
        weight1(2)=0.652145154862546;  
        weight1(3)=weight1(2);  
        weight1(4)=weight1(1);  
%  
    else                 % 5-point quadrature rule  
        point1(1)=-0.906179845938664;  
        point1(2)=-0.538469310105683;  
        point1(3)=0.0;  
        point1(4)=-point1(2);  
        point1(5)=-point1(1);
```

```
weight1(1)=0.236926885056189;  
weight1(2)=0.478628670499366;  
weight1(3)=0.568888888888889;  
weight1(4)=weight1(2);  
weight1(5)=weight1(1);  
%  
end  
%  
%----- end of feglqdl.m -----
```

P. feglqd2

```
%----- feglqd2.m -----
%
function [point2,weight2]=feglqd2(nglx,ngly)
%-----
% Purpose:
%     determine the integration points and weighting coefficients
%     of Gauss-Legendre quadrature for two-dimensional integration
%
% Synopsis:
%     [point2,weight2]=feglqd2(nglx,ngly)
%
% Variable Description:
%     nglx - number of integration points in the x-axis
%     ngly - number of integration points in the y-axis
%     point2 - vector containing integration points
%     weight2 - vector containing weighting coefficients
%-----
%
% determine the largest one between nglx and ngly
%
    if nglx > ngly
        ngl=nglx;
    else
        ngl=ngly;
    end
%
% initialization
%
    point2=zeros(ngl,2);
    weight2=zeros(ngl,2);
%
% find corresponding integration points and weights
%
    [pointx,weightx]=feglqdl(nglx);    % quadrature rule for x-axis
    [pointy,weighty]=feglqdl(ngly);    % quadrature rule for y-axis
%
% quadrature for two-dimension
%
    for intx=1:nglx                    % quadrature in x-axis
        point2(intx,1)=pointx(intx);
        weight2(intx,1)=weightx(intx);
    end
%
    for inty=1:ngly                    % quadrature in y-axis
        point2(inty,2)=pointy(inty);
        weight2(inty,2)=weighty(inty);
    end
%
%----- end of feglqd2.m -----
```

Q. glqd2

```
%----- feglqd2.m -----
%
function [point2,weight2]=glqd2(nglx,ngly)
%-----
% Purpose:
%     determine the integration points and weighting coefficients
%     of Gauss-Legendre quadrature for two-dimensional integration
%
% Synopsis:
%     [point2,weight2]=feglqd2(nglx,ngly)
%
% Variable Description:
%     nglx - number of integration points in the x-axis
%     ngly - number of integration points in the y-axis
%     point2 - vector containing integration points
%     weight2 - vector containing weighting coefficients
%-----
%
% determine the largest one between nglx and ngly
%
    if nglx > ngly
        ngl=nglx;
    else
        ngl=ngly;
    end
%
% initialization
%
    point2=zeros(ngl,2);
    weight2=zeros(ngl,2);
%
% find corresponding integration points and weights
%
    [pointx,weightx]=feglqdl(nglx);    % quadrature rule for x-axis
    [pointy,weighty]=feglqdl(ngly);    % quadrature rule for y-axis
%
% quadrature for two-dimension
%
    for intx=1:nglx                    % quadrature in x-axis
        point2(intx,1)=pointx(intx);
        weight2(intx,1)=weightx(intx);
    end
%
    for inty=1:ngly                    % quadrature in y-axis
        point2(inty,2)=pointy(inty);
        weight2(inty,2)=weighty(inty);
    end
%
%----- end of glqd2.m -----
```

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Prof. Joshua Gordis -----(1)
Naval Postgraduate School
Monterey, California
4. Dr. Jung-Kyu Ryou -----(3)
Agency for Defense Development
Yuseong P.O. Box 35, Daejeon, 305-600, Rep. of Korea